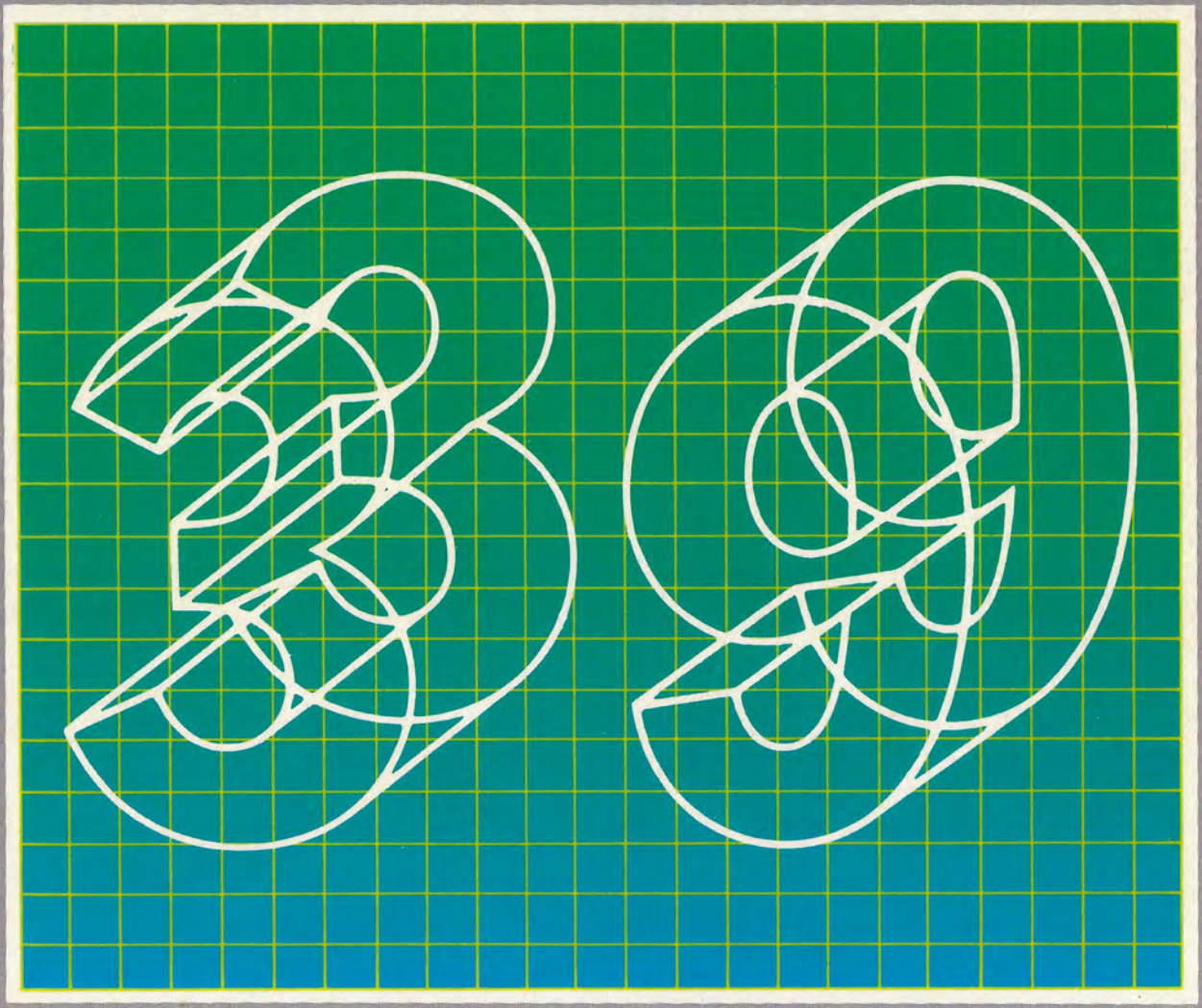
BIBLIOTECA PRACTICA

TALLER DE INFORMATICA

TRUCOS - SPRITES
BRICOLAGE DEL HARD
AULA ABIERTA
LOGO - TERMINOLOGIA

PROGRAMAS VALIDOS PARA AMSTRAD, IBM, SPECTRUM, COMMODORE Y MSX



PON A TRABAJAR A TU ORDENADOR

HAZ COSAS UTILES CON EL

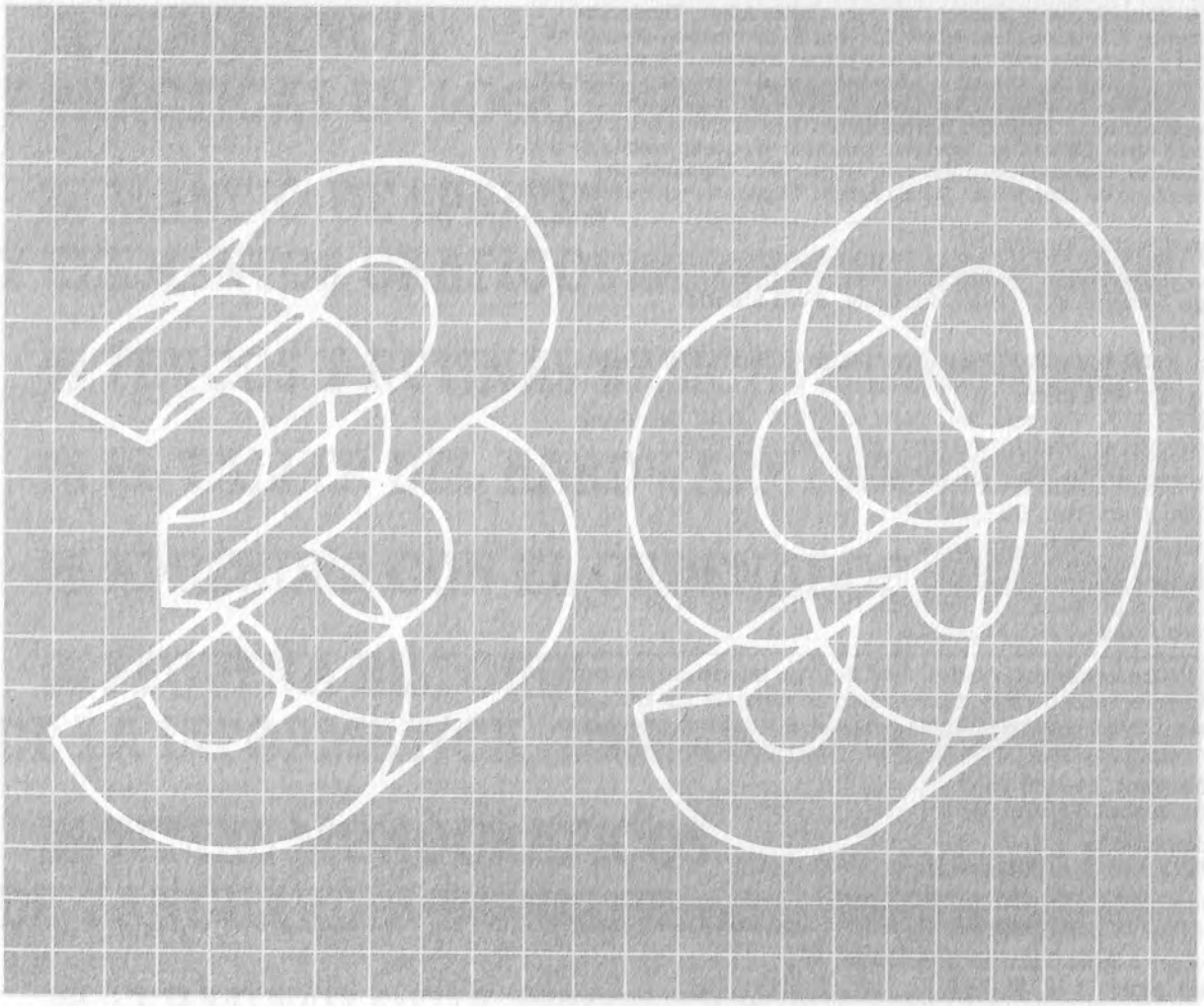
EDICIONES SIGLO CULTURAL

BIBLIOTECA PRACTICA

TALLER DE INFORMATICA

TRUCOS - SPRITES
BRICOLAGE DEL HARD
AULA ABIERTA
LOGO - TERMINOLOGIA

PROGRAMAS VALIDOS PARA AMSTRAD, IBM, SPECTRUM, COMMODORE Y MSX



EDICIONES SIGLO CULTURAL

Una publicación de

EDICIONES SIGLO CULTURAL, S. A.

Director-editor:
RICARDO ESPAÑOL CRESPO.

Gerente:
ANTONIO G. CUERPO.

Directora de producción:
MARIA LUISA SUAREZ PEREZ.

Director de la colección:
JOSE ARTECHE, Ingeniero de Telecomunicación

Diseño:
BRAVO-LOFISH.

Maquetación:
D. S. M.

Dibujos:
JOSE OCHOA

Tomo 39. «Experiencias y prácticas en logo», Carlos Albert, Técnico de Infomática; Adoración Llena Jubelo, Técnica de Informática. «Manejo de sprites y elementos gráficos, «Trucos y rutinas básicas», Francisco Morales. Técnico de Informática. «Aprende con el ordenador», AULA DE INFORMATICA APLICADA: José María Esteban, Diplomado en Telecomunicación; Alejandro Marcos, Licenciado en Química, «El Taller de Hardware», Carlos Rey, Ingeniero Industrial. «Pequeña historia de la Informática», «Temas monográficos de vanguardia», «Terminología», «Vocabulario de Informática», Blanca Arbizu, Técnico de Informática.

Ediciones Siglo Cultural, S. A.

Dirección, redacción y administración:
Pedro Teixeira, 8-2.°, Teléf. 455 09 99. 28020 Madrid.

Publicidad:
Gofar Publicidad, S. A. Benito de Castro, 12 bis. 28028 Madrid.

Distribución en España:
COEDIS, S. A. Valencia, 245. Teléf. 215 70 97. 08007 Barcelona.
Delegación en Madrid: Serrano, 165. Teléf. 411 11 48.

Distribución en Ecuador: Muñoz Hnos.

Distribución en Perú: DISELPESA.

Distribución en Chile: Alfa Ltda.

Importador exclusivo Cono Sur:
CADE, S.R.L. Pasaje Sud América, 1532. Teléf.: 21 24 64.
Buenos Aires - 1.290. Argentina.

Todos los derechos reservados. Este libro no puede ser, en parte o totalmente, reproducido, memorizado en sistemas de archivo, o transmitido en cualquier forma o medio, electrónico, mecánico, fotocopia o cualquier otro, sin la previa autorización del editor.

ISBN del tomo: 84-7688-180-0
ISBN de la obra: 84-7688-047-2

Fotocomposición:
ARTECOMP, S. A. Albarracín, 50. 28037 Madrid.

Imprime:
GRAFICAS EMA. (Madrid)

© Ediciones Siglo Cultural, S. A., 1986

Depósito legal: M-43.185-1986
Printed in Spain - Impreso en España.

Suscripciones y números atrasados:
Ediciones Siglo Cultural, S. A.
Pedro Teixeira, 8-2.°. Teléf. 259 73 31. 28020 Madrid.

Enero, 1988

P.V.P. Canarias: 355.—Ptas.

INDICE

EXPERIENCIA Y PRACTICAS EN LOGO 5

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS 13

TRUCOS Y RUTINAS BASICAS 19

EL TALLER DEL HARDWARE 25

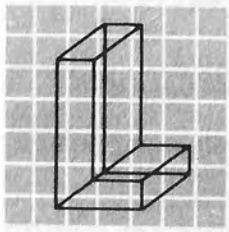
APRENDER CON EL ORDENADOR 33

PEQUEÑA HISTORIA DE LA INFORMATICA 43

TEMAS MONOGRAFICOS DE VANGUARDIA 49

TERMINOLOGIA 55

Música para la versión de Logo de los MSX



A versión de Logo para los ordenadores MSX dispone de dos primitivas para generar diferentes sonidos y poder crear piezas musicales con ellas.

Se trata de:

— SONIDO n1 n2
— INTERPRETA palabra
— INTERPRETA lista

Vamos a estudiarlas detenidamente.

— SONIDO n1 n2

Para estudiar esta primitiva tenemos que conocer de antemano una serie de cosas.

Estos ordenadores utilizan una serie de registros de los que dispone para generar los sonidos. Son catorce registros y están codificados del 0 al 13.

También disponen de tres canales de sonido, el canal A, el B y el C.

Cada registro o registros se encarga de gestionar las diferentes funciones de los canales:

— Los registros 0 y 1 son los que se encargan de controlar la frecuencia del canal A.

— Los registros 2 y 3 son los que se encargan de controlar la frecuencia del canal B.

— Los registros 4 y 5 son los que se encargan de controlar la frecuencia del canal C.

— El registro 6 se encarga de controlar la frecuencia del ruido.

— El registro 7 selecciona un canal para la generación de tonos y ruidos.

— El registro 8 controla el volumen del canal A.

— El registro 9 controla el volumen del canal B.

— El registro 10 controla el volumen del canal C.

— Los registros 11 y 12 controlan la frecuencia de variación de volumen.

— El registro 13 selecciona la variación de volumen.

A cada uno de estos registros se le puede asignar un valor, que debe estar comprendido entre los siguientes:

N.° de registro	Valores
0	0 a 255
1	0 a 15
2	0 a 255
3	0 a 15
4	0 a 255
5	0 a 15
6	0 a 31
7	0 a 63
8	0 a 15
9	0 a 15
10	0 a 15
11	0 a 255
12	0 a 255
13	0 a 14

Conocidos estos datos, la primitiva SONIDO necesita dos valores, n1 y n2:

— n1 será el número del registro.
— n2 será el valor asignado al registro.

Algunas versiones de Logo me permiten tener tres canales simultáneos para interpretar sonidos.

— INTERPRETA palabra
— INTERPRETA lista

La primitiva INTERPRETA acepta como dato una palabra o una lista.

La palabra que se especifica ha de ser una cadena de caracteres especiales que representan valores musicales. Estos valores son:

Tn°— Ha de ser un número comprendido entre 32 y 255. Representa el tiempo que dura una nota. Inicialmente el valor de T es 120.

On°— Ha de ser un número comprendido entre 1 y 8. Representa la octava. Inicialmente su valor es 4.

Ln°— Ha de ser un número comprendido entre 1 y 64. Representa la duración o longitud de una nota. Inicialmente su valor es 4.

Nn°— Ha de ser un número comprendido entre 0 y 96. Representa el valor de una nota numérica. La octava 1 contiene las notas 1 a 12, la octava 2 las notas 13 a 24, y así sucesivamente hasta llegar a la octava 8, que contendrá las notas 85 a 96.

DO-RE-MI-FA-SO-LA-SI — Representa una nota musical. Si a continuación de la nota se expresa:

— Un número comprendido entre 1 y 64, indicará la duración de la nota.

— El carácter *b* representará el bemol de la nota.

— El carácter :, la duración de la nota se incrementará en un medio.

— Si no se especifica nada, se tomará el valor de la nota.

Observa que la nota SOL ha de escribirse como SO por admitir únicamente dos caracteres.

PAn°— Ha de ser un número comprendido entre 1 y 64. Representa una pausa cuya duración la determina el número.

Vn°— Ha de ser un número comprendido entre 0 y 15. Determina el volumen de la voz.

Si el dato que especificamos para la primitiva INTEPRETA es una palabra, se interpretará por el canal 1.

Los caracteres que se especifican en la palabra han de ir escritos en letras mayúsculas.

Si se desea indicar una voz nula se hará mediante la palabra vacía (").

Si el dato que especificamos en la primitiva INTERPRETA es una lista, ésta podrá estar formada por un máximo de tres palabras y cada una de ellas será interpretada por cada uno de los tres canales que existen.

Si introducimos:

? INTERPRETA "O2L10DO"

Sonará la nota DO de la octava segunda y con una duración de 10.

Si introducimos:

? 'INTERPRETA "O5DOREMIFASOLASI"

la nota DO se oirá simultáneamente en las tres octavas que se especifican.

? 'INTERPRETA "O5DOREMIFASOLASI"

Nos permitirá oír la escala completa en la octava quinta.

?INTERPRETA [O5DOREMIFASOLASI O6DOREMIFASOLASI O7DOREMIFASOLASI]

Interpretará la escala completa en las tres octavas que se especifican y la vez.

Música para la versión de Logo del SPECTRUM

La versión de Logo para el ordenador SPECTRUM dispone de una sola primitiva con la que podemos generar sonidos.

Se trata de la primitiva:

? SONIDO n m

Como puedes observar, esta primitiva utiliza dos parámetros para su funcionamiento, *n* y *m*.

El parámetro *n* es el que va a indicar el tiempo de duración de la nota.

Puedo indicar la duración de una nota. Utiliza las hojas milimetradas para diseñar un dibujo.

El valor de *n* ha de estar comprendido entre 0 y 255.

La duración del sonido viene dado en segundos.

El parámetro *m* es el que indica la frecuencia de la nota que se desea hacer sonar.

El valor de *m* ha de estar comprendido entre -62 y 75.

Música para la versión de Logo del AMSTRAD

La versión de Logo para el AMSTRAD cuenta con la primitiva SONIDO para realizar sonidos con el ordenador.

Esta primitiva necesita cuatro parámetros para su funcionamiento, que son:

— **Primer parámetro:**

Indica el número del canal por el cual se desea que sea interpretado el sonido.

Existen tres canales y están numerados del 1 al 3.

— **Segundo parámetro:**

Indica el tono de la nota. La nota sonará más aguda cuanto mayor sea el valor de este parámetro.

— **Tercer parámetro:**

Indica la duración que va a tener la nota.

— **Cuarto parámetro:**

Indica el volumen de la nota.

Los valores de este parámetro han de estar comprendidos entre 1 y 15.

Los parámetros que se especifican han de estar encerrados entre corchetes:

? SONIDO [p1 p2 p3 p4]

Cuadro resumen

— SONIDO n1 n2 (para los MSX)

Asigna al registro del generador de sonidos que se especifica en n1 el valor que se especifica en n2.

— INTERPRETA palabra (para los MSX)

INTERPRETA lista

Interpreta la cadena de caracteres que forman la palabra en el generador de sonidos.

Si sólo se especifica una palabra, la interpretación se hará por el canal 1. Si especifica una lista, ésta no podrá contener más de tres palabras y se interpretarán simultáneamente por los canales 1, 2 y 3.

— SONIDO n m (para el Spectrum)

Genera un sonido de duración *n* y de frecuencia *m*.

— SONIDO [p1 p2 p3 p4] (para el AMSTRAD)

Genera un sonido el canal que especifica el parámetro p1, en el tono que indica el parámetro p2, con la duración que especifica el parámetro p3 y el volumen que indica el parámetro p4.

Ejercicios

1. *Consigue con el Logo el menú diario.*
2. *Intenta dibujar la paleta de un pintor.*
3. *Interpreta la siguiente partitura con el Logo utilizando la primitiva INTERPRETA:*

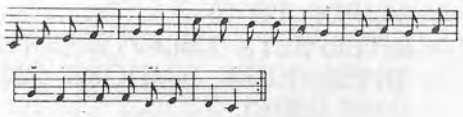


Fig. 1.

4. *Intenta realizar los siguientes dibujos:*

No olvides que los caracteres que forman las palabras de la primitiva INTERPRETA han de escribirse en mayúsculas.

a)

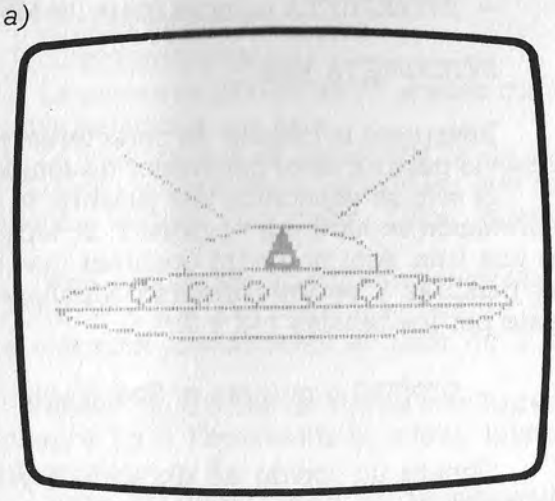


Fig. 2.

b)

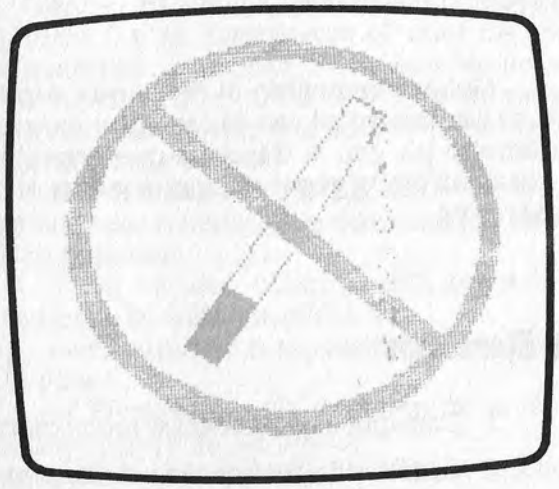


Fig. 3.

5. ¿Son correctas las siguientes órdenes?:

— SONIDO 122 10
— INTERPRETA "O5L4V15FAPA3DO
— INTERPRETA [O4DORE O2L10RE O1RE O5RE]
— ? HAZ "C 30
? HAZ "D 100
? PONESCALA [:C :D]
— ? PARA P1
> BL
> HAZ "A 5
> REPITE 36 [AV :A GD 10]
> SL AV 10
> P1 :A
> FIN

Solución a los ejercicios

1:

Lo primero que hacemos es asignar a tres variables (P1, P2 y P3) distintos primeros platos, segundos platos y postres, respectivamente.

```
? PARA PLATOS
> HAZ "P1 [LENTEJAS ARROZ SOPA
  MACARRONES]
> HAZ "P2 [POLLO TRUCHA CORDE-
  RO SALCHICHAS]
> HAZ "P3 [NARANJAS TARTA UVA
  NATILLAS]
> FIN
```

Para que el menú final no sea siempre el mismo, generamos tres números aleatorios, que determinarán el elemento de la lista que se elige.

```
? PARA ELIGE
>  HAZ "A (AZAR 4) + 1
>  HAZ "B (AZAR 4) + 1
>  HAZ "C (AZAR 4) + 1
>  FIN
```

El procedimiento SALIDA, un elemento de cada uno de las listas, dependiendo del número aleatorio anteriormente generado, y presenta en pantalla el menú del día.

```
? PARA SALIDA
>  HAZ "PF1 ELEMENTO :A :P1
>  HAZ "PF2 ELEMENTO :B :P2
>  HAZ "PF3 ELEMENTO :C :P3
>  ESCRIBE [MENU DEL DIA]
>  PONCURSOR LISTA 0 4
>  ES [PRIMER PLATO.....]
>  PONCURSOR LISTA 18 4
>  ES :PF1
>  ES [SEGUNDO PLATO....]
>  PONCURSOR LISTA 18 5
>  ES :PF2
>  PONCURSOR LISTA 0 7
>  ES [POSTRE..........]
>  PONCURSOR LISTA 18 7
>  ES :PF3
>  FIN
```

Combina los dibujos y la música.

Todos los procedimientos anteriores son ejecutados uno detrás de otro gracias al procedimiento MENU.

```
?PARA MENU
> PT
> BT
> PLATOS
> ELIGE
> SALIDA
> PONCURSOR LISTA 0 20
> FIN
```

Veamos qué menú tenemos para hoy:

? MENU

Fig. 4.

Si lo ejecutamos de nuevo:

? MENU

Fig. 5.

2:

Dibujamos la paleta con los siguientes procedimientos.

```
? PARA FORMA
> BP
> PONCL 1
> PONESCALA [100 65]
> SL
> PONPOS [-60 10]
> BL
> REPITE 30 [AV 14 GD 10]
> GD 90
> REPITE 14 [AV 7 GI 10]
> GD 80 AV 3
> SL
> CENTRO
> BL
> RELLENA
> FIN
```

El procedimiento FORMA dibuja la forma de la paleta y la rellena de color.

```
? PARA COLORES
> PONESCALA [100 79]
> SL
> PONPOS [-40 25]
> PONCL 2
> CIR
> SL PONX -30
> BL RELLENA
> SL
> PONPOS [-10 45]
> PONCL 3
> CIR
> SL PONX 0
> BL RELLENA
> SL
> PONPOS [30 45]
> PONCL 2
> CIR
> SL PONX 40
> BL RELLENA
> SL
> PONPOS [60 20]
> PONCL 3
> CIR
> SL PONX 70
> BL RELLENA
> SL
> PONPOS [55 -15]
> PONCL 4
```

```
> CIR
> SL PONX 60
> BL RELLENA
> SL
> PONPOS [20 -30]
> PONCL 3
> CIR
> SL PONX 30
> BL RELLENA
> SL
> PONPOS [-10 0]
> PONCL 0
> CIR
> SL PONX 0
> BL RELLENA
> FIN
```

El procedimiento anterior dibuja y rellena de color varios círculos dentro de la paleta.

```
? PARA PINCELES
> SL
> PONCL 2
> PONPOS [0 -5]
> BL
> PONRUMBO 350
> AV 120
> GD 90
> AV 5
> GD 92
> AV 120
> PONRUMBO 10
> AV 120
> GD 90
> AV 5
> GD 92
> AV 120
> FIN
```

PINCELES dibuja dos pinceles sobre la paleta.

```
? PARA PAL
> FORMA
> COLORES
> PINCELES
> FIN
```

Y finalmente PAL se encarga de llamar a cada uno de ellos en el orden correspondiente.

Para conseguir el dibujo teclea:

?PAL

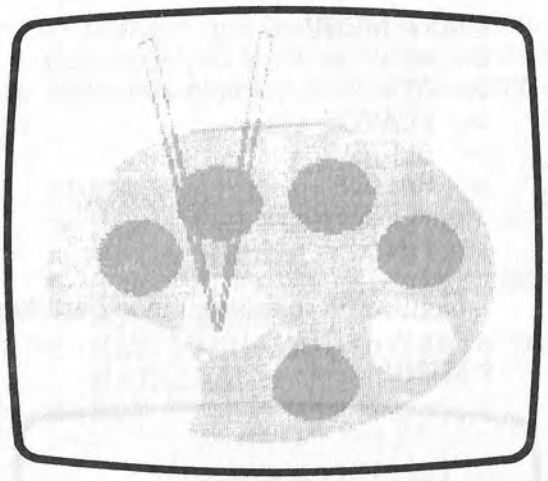


Fig. 6.

3:

En primer lugar, obtenemos las notas por las que está compuesta la partitura:

DO RE MI FA
SOL SOL
DO DO (EN UNA OCTAVA SUPERIOR)
SI SI
LA SOL
SOL LA SOL LA
SOL FA
FA FA RE MI
RE DO

Establecemos el tiempo de duración de cada nota. Al intervenir sólo negras y corcheas, establecemos los siguientes tiempos de duración:

negra = 16
corchea = 8

Observa que en este caso la duración más larga (negra) se representa con un número más alto que la duración más corta (corchea).

A continuación definimos un procedimiento para interpretar esta pieza musical.

Partimos de la octava quinta como base:

```
? PARA CANCION
? INTERPRETA "O5
```

Puedes regular el volumen de la voz.

```
? INTERPRETA "L16
? INTERPRETA "DORE
? INTERPRETA "MIFA
? INTERPRETA "L8
? INTERPRETA "SOSO
? INTERPRETA "O6
? INTERPRETA "L16
? INTERPRETA "DODO
? INTERPRETA "O5
? INTERPRETA "SISI
? INTERPRETA "L8
? INTERPRETA "LASO
? INTERPRETA "L16
? INTERPRETA "SOLA
? INTERPRETA "SOLA
? INTERPRETA "L8
? INTERPRETA "SOFA
? INTERPRETA "L16
? INTERPRETA "FAFA
? INTERPRETA "REMI
? INTERPRETA "L8
? INTERPRETA "REDO
? FIN
```

Hemos separado la cadena para que observes todos los cambios que hay, tanto en octavas como en la longitud de cada nota.

Para escuchar esta pieza introduce:

```
? CANCION
```

4:

a) El dibujo del platillo lo realizamos con el siguiente procedimiento:

```
? PARA PLA
> OT
> PONESCALA [100 80]
> PONCL 1
> BP
> SL
> PONPOS [-60 0]
> BL
> PONESCALA [150 20]
> REPITE 36 [AV 7 GD 10]
> PONRUMBO 90
> AV 80
> GI 90
> REPITE 7 [AV 7 GI 10]
> PONRUMBO 0
> PONESCALA [150 60]
> REPITE 19 [AV 3 GI 10]
> SL
> PONPOS [10 35]
> PONRUMBO 30
> BL AV 30
> SL
> PONPOS [-10 35]
> PONRUMBO 325
> BL AV 30
> SL
> PONPOS [-35 10]
> PONRUMBO 90
> PONESCALA [100 80]
> REPITE 6 [SL AV 15 BL REPITE 36
  [AV 0.5 GD 10]]
> SL
> PONPOS [0 12]
> PONRUMBO 0
> PONCL 2
> MT
> FIN
```

Para ver el resultado teclea:

```
? PLA
```

b) La señal de prohibido fumar la realizamos con los procedimientos SEÑAL Y HUMO.

```
? PARA SEÑAL
> BP
> PONCL 2
> SL
> PONPOS [-60 0]
> PONRUMBO 0
> BL
> REPITE 36 [AV 10 GD 10]
> SL
> GD 90
> AV 5 GI 90
> BL
> REPITE 40 [AV 9 GD 10]
> SL GI 90 AV 3
> BL RELLENA
> GD 90 AV 3
> PONRUMBO 135
> AV 111
> PONRUMBO 0
> AV 10
> PONRUMBO 315
```

Consigue efectos especiales con los colores y la música.

```
> AV 103
> SL
> PONRUMBO 150
> AV 5
> BL RELLENA
> SL
> PONCL 3
> PONPOS [25 35]
> PONRUMBO 215
> BL AV 40
> SL AV 10
> BL AV 35
> GD 90 AV 10
> GD 90 AV 37
> SL AV 10
> BL AV 18
> GD 90 AV 10
> RE 10 GI 90
> AV 20 GD 90
> AV 10
> PONRUMBO 240
> SL AV 10
> BL RELLENA
> REPITE 50 [HUMO]
> FIN
```

Para dibujar el humo el procedimiento SEÑAL llama 50 veces al procedimiento HUMO

```
? PARA HUMO
> HAZ "X AZAR 5
> HAZ "X -32 + :X
> HAZ "Y AZAR 80
> HAZ "Y -30 + :Y
> PUNTO LISTA :X :Y
> FIN
```

Para ver cómo queda teclea:

```
?SEÑAL
```

5:

```
— SONIDO 122 10
INCORRECTO
```

Se da como número de registro el 122, el cual no existe.

```
— INTERPRETA "O5L4V15FAPA3DO
CORRECTO
```

Sonará la nota FA en la octava cinco durante un tiempo de cuatro y con un volumen de voz igual a 15. A continuación habrá una pausa de 3 y sonará la nota DO.

```
— INTERPRETA [O4DORE O2L1ORE
  O1RE O5RE]
  INCORRECTO
```

El máximo de palabras que se pueden expresar en una lista para la primitiva INTERPRETA es de tres.

```
? HAZ "C 30
? HAZ "D 100
? PONESCALA [:C :D]
INCORRECTO
```

El Logo devuelve el mensaje de error:

PONESCALA NO ACEPTA :C COMO DATO

Si quieres utilizar variable con la primitiva PONESCALA hazlo así:

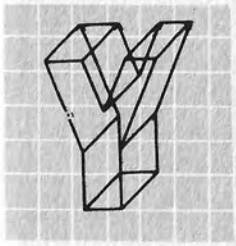
```
PONESCALA LISTA :C :D
? PARA P1
> BL
> HAZ "A 5
> REPITE 36 [AV :A GD 10]
> SL AV 10
> P1 :A
> FIN

INCORRECTO
```

Aunque al ejecutarlo el Logo no devuelva ningún error, la llamada recursiva no es correcta, el procedimiento P1 no tiene ninguna entrada.

Controla la intensidad del altavoz. La interpretación de sonidos varía según el ordenador con el que estemos trabajando.

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS



para terminar con el MSX, al igual que vamos a hacer con el resto de los ordenadores, vamos a ver un programa que conjuga la alta resolución con los SPRITES y con el diseño de caracteres. Este programa, una versión sencilla y reducida del famoso BREAK OUT, nos permitirá no sólo pasar unos momentos agradables, sino aprender algo más sobre la realización de juegos, sobre el movimiento, sobre la definición de caracteres y el uso de SPRITES.

```
1000 REM ***************************
1010 REM *                         *
1020 REM *   B R E A K    O U T    *
1030 REM *                         *
1040 REM ***************************
1050 REM
1060 REM ****************************
1070 REM *                          *
1080 REM * (c) Ed. Siglo Cultural.  *
1090 REM *                          *
1100 REM * (c) 1987.                *
1110 REM *                          *
1120 REM ****************************
1130 REM
1140 REM ******************
1150 REM * INICIALIZACION *
1160 REM ******************
1170 REM
1180 LET R=2000
1190 DIM L(32,24)
1200 SCREEN 1,0
1210 KEY OFF
1220 WIDTH 32
1230 RESTORE 3010
1240 GOSUB 3250
1250 LET S=1
1260 LET ST=1
1270 REM
1280 REM ***************************
1290 REM * LECTURA DE LOS SPRITES *
1300 REM ***************************
1310 REM
```

```
1320 RESTORE 1740
1330 FOR A=0 TO 2
1340    FOR B=0 TO 7
1350       READ C
1360       LET A$=A$+CHR$(C)
1370    NEXT B
1380    SPRITE$(A)=A$
1390    LET A$=""
1400 NEXT A
1410 REM
1420 REM ******************************************
1430 REM *                                        *
1440 REM *   P R O G R A M A   P R I N C I P A L  *
1450 REM *                                        *
1460 REM ******************************************
1470 GOSUB 3660
1480 LET V=3
1490 LET P=0
1500 LET SC=1
1510 LET LD=144
1520 GOSUB 1960
1530 GOSUB 2140
1540 GOSUB 3320
1550 GOSUB 3460
1560 LET XP=131
1570 LET XB=131
1580 LET IX=-3
1590 LET YP=165
1600 LET IY=-1
1610 GOSUB 2440
1620 GOSUB 2520
1630 LET YB=157
1640 GOSUB 1810
1650 ON INTERVAL=1 GOSUB 2240
1660 INTERVAL ON
1670 GOTO 1670
1680 END
1690 REM
1700 REM *****************************************
1710 REM * DATAS CON LA DEFINICION DE LOS SPRITES *
1720 REM *****************************************
1730 REM
1740 DATA 0,63,64,95,95,64,63,0
1750 DATA 0,252,2,250,250,2,252,0
1760 DATA 0,60,126,126,126,126,60,0
1770 REM
1780 REM **************************
1790 REM * IMPRESION DE LA PALETA *
1800 REM **************************
1810 REM
1820 PUT SPRITE 0,(XP,YP),10,0
1830 PUT SPRITE 1,(XP+8,YP),10,1
1840 RETURN
1850 REM
1860 REM ********************
1870 REM * USO DEL JOYSTICK *
1880 REM ********************
1890 REM
1900 LET J=STICK(ST)
```

```
1910 IF J=3 THEN IF XP<207 THEN LET XP=XP+4
1920 IF J=7 THEN IF XP>28 THEN LET XP=XP-4
1930 RETURN
1940 REM
1950 REM *****************
1960 REM * PONE RECUADRO *
1970 REM *****************
1980 REM
1990 FOR A=3 TO 28
2000    LOCATE A,0
2010    PRINT "O"
2020    LOCATE A,22
2030    PRINT "O"
2040 NEXT A
2050 FOR A=0 TO 32
2060    LOCATE 3,A
2070    PRINT "O"
2080    LOCATE 28,A
2090    PRINT "O"
2100 NEXT A
2110 RETURN
2120 REM
2130 REM ***************************
2140 REM * IMPRIME MENSAJE PUNTOS *
2150 REM ***************************
2160 REM
2170 LOCATE 4,23
2180 PRINT "PUNTOS : ";
2190 LOCATE 20,23
2200 PRINT "VIDAS : ";
2210 RETURN
2220 REM
2230 REM ************************
2240 REM * IMPRESION DE LA BOLA *
2250 REM ************************
2260 REM
2270 IF XB>219 OR XB<26 THEN LET IX=-IX:PLAY "O3L64A"
2280 IF YB<8 THEN LET IY=1:PLY "O3L64A"
2290 IF YB>160 THEN GOSUB 2380:GOTO 1560
2300 LET XB=XB+IX
2310 LET YB=YB+IY
2320 PUT SPRITE 2,(XB,YB),15,2
2330 GOSUB 1870
2340 GOSUB 2770
2350 RETURN
2360 REM
2370 REM ********************************
2380 REM * PERDIDA DE UNA DE LAS VIDAS *
2390 REM ********************************
2400 REM
2410 IF XB>XP-2 AND XB<XP+18 THEN GOSUB 2930:RETURN 2300
2420 PLAY "O3L3CR9CL8CL3CL4R8D+R16L6DL3DL6CL4CL6R16O2BO3L3C"
2430 LET V=V-1
2440 LOCATE 27,24
2450 PRINT V;"  ";
2460 FOR I=1 TO 1500
2470 NEXT I
2480 IF V=0 THEN GOTO 2600
2490 RETURN
```

```
2500 REM
2510 REM **********************
2520 REM * IMPRIME LOS PUNTOS *
2530 REM **********************
2540 REM
2550 LOCATE 12,23
2560 PRINT P;"    ";
2570 RETURN
2580 REM
2590 REM *********************
2600 REM * FIN DE LA PARTIDA *
2610 REM *********************
2620 REM
2630 SCREEN 0
2640 IF R>P THEN LET R=P
2650 LOCATE 10,10
2660 PRINT "OTRA PARTIDA ? (S/N)"
2670 LOCATE 12,5
2680 PRINT "RECORD : ";R
2690 LET A$=INPUT$(1)
2700 IF A$="S" OR A$="s" THEN RUN
2710 IF A$<>"N" AND A$<>"n" THEN GOTO 2690
2720 LOCATE 0,20
2730 PRINT "ADIOS ..."
2740 END
2750 REM
2760 REM ****************************************
2770 REM * CHOQUE DE LA PELOTA CON EL LADRILLO *
2780 REM ****************************************
2790 REM
2800 IF L((XB+4)/8,(YB+4)/8)=1 THEN LET L((XB+4)/8,(YB+4)/8)=0 ELSE RETURN
2810 PLAY "O7L64F"
2820 LOCATE (XB+4)/8,(YB+4)/8
2830 PRINT " "
2840 LET P=P+10*SC
2850 GOSUB 2520
2860 LET LD=LD-1
2870 IF LD=0 THEN GOSUB 3570
2880 LET IY=-IY
2890 IF RND(-TIME)*10>5 THEN LET IY=-IY:LET IX=-IX
2900 RETURN
2910 REM
2920 REM **********
2930 REM * CHOQUE *
2940 REM **********
2950 REM
2960 LET IX=(NT((XB-XP+8)/4)/2)
2970 LET IY=-1
2980 RETURN
2990 REM
3000 REM ******************
3010 REM * PANTALLA No. 1 *
3020 REM ******************
3030 REM
3040 DATA 0,127,64,64,64,64,127,0
3050 DATA 0,254,2,2,2,2,254,0
3060 REM
3070 REM ******************
3080 REM * PANTALLA No. 2 *
```

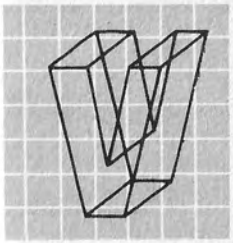
```
3090 REM ******************
3100 REM
3110 DATA 0,127,100,91,91,100,127,0
3120 DATA 0,254,146,110,110,146,254,0
3130 REM
3140 REM ******************
3150 REM * PANTALLA No. 3 *
3160 REM ******************
3170 REM
3180 DATA 0,127,63,31,31,63,127,0
3190 DATA 0,254,252,248,248,252,254,0
3200 REM
3210 REM ****************************
3220 REM * LECTURA DE LOS LADRILLOS *
3230 REM ****************************
3240 REM
3250 FOR A=776 TO 823
3260    READ B
3270    VPOKE (BASE(7)+A),B
3280 NEXT A
3290 RETURN
3300 REM
3310 REM **********************
3320 REM * IMPRIME LADRILLOS *
     EM ********************
     EM
     FOR A=4 TO 26 STEP 2
3360    FOR B=5 TO 10
3370       LOCATE A,B
3380       IF S=1 THEN PRINT "ab"
3390       IF S=2 THEN PRINT "cd"
3400       IF S=3 THEN PRINT "ef"
3410    NEXT B
3420 NEXT A
3430 RETURN
3440 REM
3450 REM ********************
3460 REM * ACTIVA LADRILLOS *
3470 REM ********************
3480 REM
3490 FOR A=4 TO 27
3500    FOR B=5 TO 10
3510       LET L(A,B)=1
3520    NEXT B
3530 NEXT A
3540 RETURN
3550 REM
3560 REM **********************
3570 REM * SIGUIENTE PANTALLA *
3580 REM **********************
3590 REM
3600 LET S=S+1
3610 IF S=4 THEN LET S=1
3620 LET LD=144
3630 GOSUB 3320
3640 GOSUB 3460
3650 RETURN
3660 REM
```

```
3670 REM ****************
3680 REM * PRESENTACION *
3690 REM ****************
3700 REM
3710 CLS
3720 LOCATE 10,10
3730 PRINT "0. CURSOR"
3740 PRINT
3750 PRINT TAB(10);"1. JOYSTICK 1"
3760 LET A$=INKEY$
3770 IF A$=""THEN GOTO 3760
3780 IF A$<>"1" AND A$<>"2" THEN GOTO 3760
3790 LET ST=VAL(A$)
3800 CLS
3810 RETURN
```

No se ha incluido una tabla de records para que el propio usuario se confeccione la suya propia. Con ello esperamos conseguir que el lector adquiera más práctica en la realización de programas y en el cuidado de los detalles de presentación.

El programa aparece totalmente comentado para que cualquiera pueda entender qué hace cada parte del programa. Por ejemplo, como el programa sólo tiene definidas tres tipos distintos de pantallas, podemos variar el programa para que tenga todas las pantallas que queramos.

Por otro lado, el programa resulta excesivamente lento para resultar un buen juego. Esto es debido a la cantidad de comentarios y a la cantidad de líneas que ocupa. Una vez introducido y entendido, el lector puede variarlo, quitando las líneas REM y compactándolo, de forma que el programa adquiera velocidad.



AMOS a ver a continuación un programa de diseño de pantalla para el AMSTRAD. Poco hay que decir sobre él, salvo que es uno de los mejores que se han escrito.

El programa se maneja de forma muy sencilla, por iconos, lo cual facilita la familiarización de este tablero de dibujo.

Empecemos con las teclas que hay que manejar en el transcurso de un dibujo:

```
1000 REM *************************
1010 REM *                       *
1020 REM *   FUNCIONES GRFICAS   *
1030 REM *  -------------------  *
1040 REM *                       *
1050 REM *************************
1060 REM
1070 REM ********************************
1080 REM *                              *
1090 REM * (c) Ediciones Siglo Cultural. *
1100 REM *                              *
1110 REM * (c) 1987.                    *
1120 REM *                              *
1130 REM ********************************
1140 REM
1150 LET A1=1
1160 LET A2=1
1170 LET X5=0
1180 LET AU=1
1190 DIM K1$(6)
1200 LET K1$(1)="T"
1210 LET K1$(2)="S"
1220 LET K1$(3)="S"
1230 LET K1$(4)="S"
1240 LET K1$(5)="N"
1250 LET K1$(6)="N"
1260 KEY OFF
1270 FOR I=1 TO 10
1280    KEY I,""
1290 NEXT I
1300 KEY 1,CHR$(13)+"GOTO 5760"+CHR$(13)
1310 SCREEN 0
1320 CLS
1330 LOCATE 10,0
1340 PRINT "G R A F I C A S"
```

```
1350 PRINT TAB(10);"==============="
1360 PRINT:PRINT:PRINT
1370 DEF FN Y(X)=SIN(X)/X
1380 PRINT:PRINT:PRINT:PRINT:PRINT
1390 PRINT SPC(8);"1...Dibujar grafica."
1400 PRINT SPC(8);"2...Escribir formula."
1410 PRINT SPC(8);"3...Definir formato."
1420 PRINT SPC(8);"4...Salir del programa."
1430 PRINT:PRINT:PRINT
1440 PRINT:PRINT:PRINT
1450 LOCATE 10,21
1460 PRINT " <<ELIGE OPCION>>"
1470 LET K$=INKEY$
1480 IF K$="" THEN GOTO 1470
1490 LET K=ASC(K$)-48
1500 IF K<1 OR K>4 THEN GOTO 1470
1510 ON K GOSUB 2320,1830,1550,1530
1520 GOTO 1310
1530 CLS
1540 END
1550 REM
1560 REM ************************
1570 REM * DEFINCION DE FORMATO *
1580 REM ************************
1590 REM
1600 CLS
1610 PRINT
1620 PRINT"  DEFINICION DEL FORMATO"
1630 PRINT"==========================="
1640 PRINT
1650 PRINT
1660 RESTORE
1670 FOR F=1 TO 6
1680    READ AF$
1690    READ NE$
1700    READ ME$
1710    PRINT ME$;
1720    IF F=6 THEN PRINT "Margen cuando XY ==> infi. (";
1730    PRINT K1$(F);")";CHR$(29);CHR$(29);
1740    LET K$=INKEY$
1750    IF K$="" THEN GOTO 1740
1760    IF ASC(K$)>90 THEN LET K$=CHR$(ASC(K$)-32)
1770    IF K$<>"" THEN IF K$=AF$ OR K$=NE$ THEN LET K1$(F)=K$
1780    IF K$<>AF$ AND K$<>NE$ THEN PRINT ELSE PRINT K1$(F)
1790 NEXT F
1800 RETURN
1810 REM
1820 REM *************************
1830 REM *** CAMBIO DE FUNCION ***
1840 REM *************************
1850 REM
1860 CLS
1870 LOCATE 5,2
1880 PRINT "Cambiar la ecuacion y pulsar F1";
1890 LOCATE 7,5
1900 LIST 1370
1910 END
1920 CLS
1930 REM
1940 REM *******************
1950 REM *** DIBUJA EJES ***
1960 REM *******************
```

```
1970 REM
1980 SCREEN 2
1990 OPEN "GRP:"AS#1
2000 REM
2010 REM *** DIBUJO EJE X ***
2020 REM
2030 IF K1$(2)="N"THEN GOTO 2150
2040 LINE(122,0)-(132,0),15
2050 LINE(122,19)-(132,19),15
2060 LINE(122,38)-(132,38),15
2070 LINE(122,57)-(132,57),15
2080 LINE(122,76)-(132,76),15
2090 LINE(0,95)-(255,95),15
2100 LINE(122,114)-(132,114),15
2110 LINE(122,133)-(132,133),15
2120 LINE(122,152)-(132,152),15
2130 LINE(122,171)-(132,171),15
2140 LINE(122,190)-(132,190),15
2150 REM
2160 REM *** DIBUJO EJE Y ***
2170 REM
2180 IF K1$(3)="N"THEN GOTO 2300
2190 LINE(31,90)-(31,100),15
2200 LINE(50,90)-(50,100),15
2210 LINE(69,90)-(69,100),15
2220 LINE(88,90)-(88,100),15
2230 LINE(107,90)-(107,100),15
2240 LINE(127,0)-(127,255),15
2250 LINE(146,90)-(146,100),15
2260 LINE(165,90)-(165,100),15
2270 LINE(184,90)-(184,100),15
2280 LINE(203,90)-(203,100),15
2290 LINE(222,90)-(222,100),15
2300 IF K1$(5)="S" THEN GOSUB 3270
2310 RETURN
2320 REM
2330 REM ************************
2340 REM *** ESCALA Y AUMENTO ***
2350 REM ************************
2360 REM
2370 CLS
2380 LOCATE 1,1
2390 PRINT "Escala del eje X:";A1
2400 LOCATE 20,1
2410 INPUT A1$
2420 IF A1$<>"" THEN LET A1=VAL(A1$)
2430 LOCATE 19,1
2440 PRINT A1;"  "
2450 LOCATE 1,2
2460 PRINT "Escala del eje Y:";A2
2470 LOCATE 20,2
2480 INPUT A2$
2490 IF A2$<>"" THEN LET A2=VAL(A2$)
2500 LOCATE 19,2
2510 PRINT A2;"  "
2520 LOCATE 1,3
2530 REM
2540 REM ************************
2550 REM *** SELECCION DE PASO ***
2560 REM ************************
2570 REM
2580 PRINT "STEP:";S
```

```
2590 LOCATE 7,3
2600 INPUT S
2610 IF K1$(4)="S" THEN LET S=1/A1:LOCATE 1,5:PRINT "STEP:Autom.";S
2620 IF S=0 THEN LET S=1/A1
2630 GOSUB 1920
2640 IF K1$(1)="T" THEN GOTO 3040
2650 GOTO 2680
2660 END
2670 REM
2680 REM ****************************
2690 REM ** TRAT.GRAF.DE FUNCIONES **
2700 REM ****************************
2710 REM
2720 REM **************************
2730 REM *** DIBUJAR CON PUNTOS ***
2740 REM **************************
2750 REM
2760 REM *** DIBUJO DE F(X) ***
2770 REM
2780 ON ERROR GOTO 3230
2790 PRESET(X5,95),15
2800 LET XPOS=X5
2810 LET YPOS=95
2820 FOR X=-20 TO 20 STEP S
2830     IF K1$(6)="N" THEN IF XPOS>255 OR XPOS<0 OR YPOS>191 OR YPOS<0 THEN GOTO
 2900
2840     LET PX=INT(X*A1*8)+127
2850     LET XPOS=PX
2860     LET PY=95-INT(FN Y(X)*A2*8)
2870     LET YPOS=PY
2880     PSET(PX,PY),11
2890 NEXT X
2900 GOTO 2910
2910 LINE(0,0)-[illegible] PE
2920 PRINT #1,"*****************"
2930 PRINT #1,"*PULSA UNA TECLA*"
2940 PRINT #1,"*****************"
2950 LET A$=INPUT$(1)
2960 GOTO 1310
2970 REM
2980 REM *************************
2990 REM *** DIBUJAR CON TRAZO ***
3000 REM *************************
3010 REM
3020 REM *** DIBUJO DE F(X) ***
3030 REM
3040 ON ERROR GOTO 3230
3050 PRESET(X5,0)
3060 LET XPOS=X5
3070 LET YPOS=0
3080 FOR X=-16 TO 16 STEP S
3090     IF K1$(6)="N" THEN IF XPOS>205 OR XPOS<-205 OR YPOS>205 OR YPOS<-205 THE
N GOTO 3150
3100     LET PX=INT(X*A1*8)+127
3110     LET XPOS=PX
3120     LET PY=95-INT(FN Y(X)*A2*8)
3130     LET YPOS=PY
3140     LINE -(PX,PY),15
3150 NEXT X
3160 GOTO 2910
3170 REM
3180 REM
```

```
3190 REM ***************************
3200 REM *** TRATAMIENTO ERRORES ***
3210 REM ***************************
3220 REM
3230 REM
3240 RESUME NEXT
3250 END
3260 REM
3270 REM *************************
3280 REM *** DIBUJA CUADRICULA ***
3290 REM *************************
3300 REM
3310 LINE(0,0)-(255,0),15
3320 LINE(0,19)-(255,19),15
3330 LINE(0,38)-(255,38),15
3340 LINE(0,57)-(255,57),15
3350 LINE(0,76)-(255,76),15
3360 LINE(0,95)-(255,95),15
3370 LINE(0,114)-(255,114),15
3380 LINE(0,133)-(255,133),15
3390 LINE(0,152)-(255,152),15
3400 LINE(0,171)-(255,171),15
3410 LINE(0,190)-(255,190),15
3420 LINE(31,0)-(31,191),15
3430 LINE(50,0)-(50,191),15
3440 LINE(69,0)-(69,191),15
3450 LINE(88,0)-(88,191),15
3460 LINE(107,0)-(107,191),15
3470 LINE(127,0)-(127,191),15
3480 LINE(146,0)-(146,191),15
3490 LINE(165,0)-(165,191),15
3500 LINE(184,0)-(184,191),15
3510 LINE(203,0)-(203,191),15
3520 LINE(222,0)-(222,191),15
3530 RETURN
3540 REM
3550 REM *******************
3560 REM *** LINEAS DATA ***
3570 REM *******************
3580 REM
3590 DATA "P","T","1...Punto o trazo. ("
3600 DATA "S","N","2...Dibuja eje x.  ("
3610 DATA "S","N","3...Dibuja eje y.  ("
3620 DATA "S","N","4...Step   Auto.   ("
3630 DATA "S","N","5...Dibuja cuadricula.         ("
3640 DATA "S","N","6..."
3650 REM
3660 REM ************************
3670 REM *** FIN DEF. FUNCION ***
3680 REM ************************
3690 REM
3700 CLS
3710 PRINT "Si has definido el formato,tendras   que volver a hacerlo."
3720 PRINT
3730 PRINT "PULSA UNA TECLA"
3740 LET A$=INPUT$(1)
3750 RUN
```

En la pantalla de ICONOS

— Las teclas cursoras realizan la exclusiva función de mover el cursor por la pantalla para seleccionar la función de ICONOS requerida.

— RETURN hace que el ordenador acepte la opción que hay bajo el cursor.

— La tecla 0 → F en el teclado de dibujo, en conjunto con las teclas cursoras, hace que el cursor de iconos vaya más rápido.

— La tecla " → M en el teclado de dibujo hace que el programa pase del menú de iconos al tablero de dibujo, o al contrario, automáticamente.

En el tablero de DIBUJO

— Las teclas del teclado de definición son las siguientes:

7 = Diagonal arriba-izquierda.
8 = Arriba.
9 = Diagonal arriba-derecha.
4 = Izquierda.
6 = Derecha.
1 = Diagonal abajo-izquierda.
2 = Abajo.
3 = Diagonal abajo-derecha.
5 = FAST. Tecla que, combinada con las teclas cursoras, da mayor rapidez al cursor de dibujo.

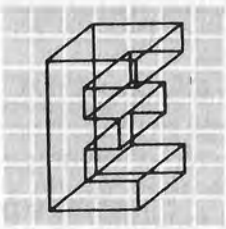
— Para seleccionar la excentricidad de la elipse, se utiliza la tecla 0 → F en el teclado de dibujo.

En el teclado de PALETA

— SPACE sirve para seleccionar el cambio de tintas del papel o de la pluma, así como para gestionar el cambio de tintas simples o compuestas.

— ALGARROBA (@) sirve para definir los tinteros con el color deseado, pudiendo ser simples o compuestos. En el último caso se pueden seleccionar, al final del proceso, la velocidad de las tintas, con las teclas cursoras.

Fuente de alimentación conmutada



EN el tomo precedente se ha presentado la posibilidad de diseñar y montar una fuente de alimentación ininterrumpida para superar la falta de constancia de la red eléctrica. Para el diseño de la fuente que suministra corriente alterna se han utilizado transistores en conmutación. En las fuentes de alimentación de la mayoría de los equipos electrónicos actuales se tiene como criterio básico de diseño la eficiencia en el consumo de energía, para lo cual es también imprescindible el empleo de técnicas de conmutación, utilizando principalmente transistores.

Por ser de interés general para el interesado en el funcionamiento de todas las partes del ordenador personal vamos a describir las fuentes de alimentación conmutada, los componentes que se utilizan y los principios básicos de operación.

Tipos de fuentes de alimentación conmutada

El principio de las fuentes de alimentación conmutadas es básicamente el suministro de corriente de forma periódica a partir de una tensión continua conseguida mediante la rectificación de la tensión de red. Mediante el control del tiempo durante el cual se suministra corriente a la carga puede conseguirse cualquier tensión de salida si se filtra convenientemente la señal pulsante generada.

El objetivo principal de suministrar la corriente «a ráfagas» radica en que el elemento de control no disipa energía, al menos teóricamente, pues o bien se encuentra en circuito abierto, intensidad nula, o bien se encuentra en corto-circuito, caída de tensión nula. Cuanto más ideal sea el comportamiento del elemento de control, tanto mayor será el rendimiento del sistema. En la práctica llegan a conseguirse rendimientos del 90%, muy por encima del límite teórico de una fuente lineal ideal.

Un segundo objetivo importante es la reducción de los componentes pesados y costosos como son los transformadores de potencia. Mediante el empleo de frecuencias elevadas de conmutación los transformadores a emplear, para conseguir el aislamiento entre la red y el equipo alimentado, resultan menos voluminosos y pesados, aunque como contrapartida requieren el empleo de componentes magnéticos de alta frecuencia, del tipo de ferritas blandas, y que tienen un coste superior al del hierro utilizado en los tranformadores normales de baja frecuencia. El balance es, sin embargo, positivo, por lo que es cada vez más frecuente encontrar en cualquier equipo fuentes conmutadas de alta frecuencia.

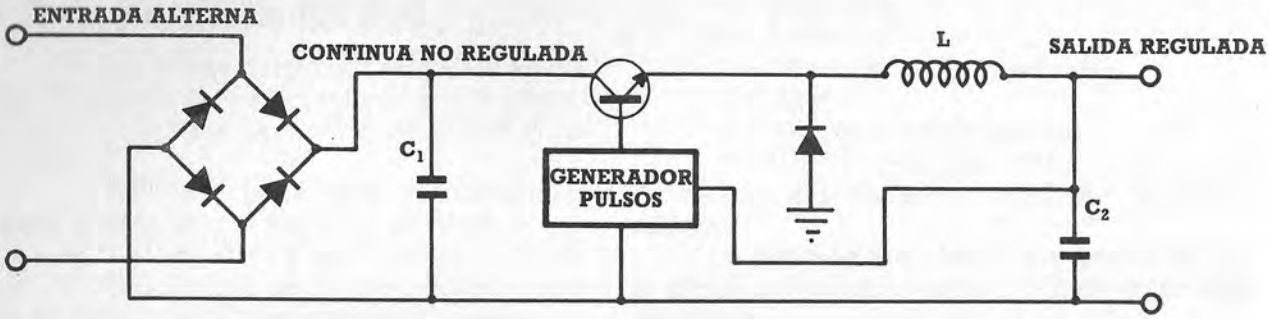


Fig. 1. Fuente conmutada reductora.

Una fuente de alimentación conmutada con tensión de salida baja, menor que la de entrada, es en esquema como se indica en la figura 1. El rectificador suministra tensión continua a partir de la red y carga los condensadores a una tensión próxima a la de pico de la entrada. El elemento conmutador, generalmente un transistor, suministra la corriente a través de una bobina a cuya salida se encuentra un condensador, en paralelo con la carga.

Al ponerse el transistor en estado de conducción la tensión en los condensadores se aplica a la bobina, que se comporta de cara a las variaciones bruscas de tensión como si fuera un circuito abierto, es decir, presenta una impedancia proporcional a la variación de intensidad. La consecuencia inmediata es que la tensión de la entrada no aparece a la salida instantáneamente, sino que comienza una subida suave, de acuerdo con los valores de los componentes.

El comportamiento transitorio del arranque del circuito presentará una subida paulatina de la tensión a la salida, hasta que se equilibre con el consumo y proporcional al tiempo de conducción del conmutador. Variando este tiempo se podrá controlar la tensión de salida dentro de ciertos márgenes.

Los circuitos típicos suelen disponer de un oscilador de frecuencia fija que determina el tiempo de comienzo de la conmutación, siendo la duración determinada por el circuito de control según las condiciones de la carga.

El acoplamiento a los circuitos de salida se realiza mediante un transformador conectado en el lugar de la carga.

En los casos en que es necesario disponer de varias tensiones de salida se utilizan reguladores lineales, a continuación de la etapa de conmutación si el consumo es reducido y diferentes arrollamientos si el consumo es de cierta importancia.

Esquema de bloques

Además del circuito de conmutación mostrado en la primera figura es necesario

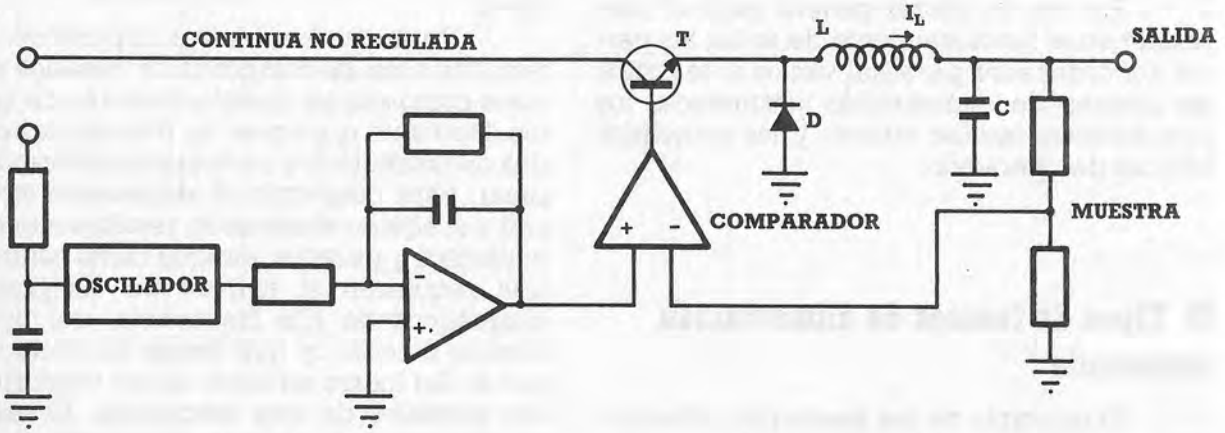


Fig. 2. Esquema de bloques de una fuente de alimentación conmutada.

considerar el circuito de control, que determina el ancho del pulso que permite conducir al transistor. Está formado por un oscilador de frecuencia fija, determinada por un circuito RC, un conversor de señal triangular y un comparador con la señal de salida.

El circuito comparador entrega a la salida unos pulsos, cuya duración en estado 1 corresponde al tiempo en que debe conducir el transistor de control. Cuando por efecto del aumento de consumo en la salida disminuye la tensión de salida, el comparador dará pulsos cuya duración en estado activo sea mayor, compensando con ello la tensión de salida y estabilizando por tanto el nivel.

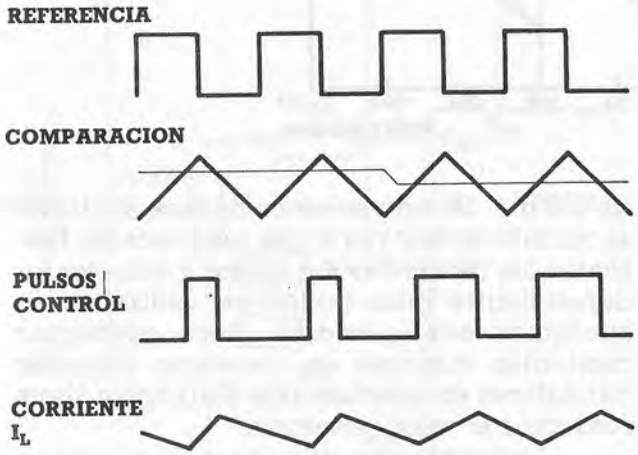


Fig. 3. Señales en el circuito.

Las señales fundamentales en el circuito son las mostradas en la figura 3. La señal de referencia es producida por el oscilador, generalmente de forma cuadrada. Mediante el integrador se convierte la señal a triangular simétrica. Al compararla con la señal de salida se generan los pulsos de ancho variable, proporcionalmente al consumo en cada instante. En la figura se muestran dos estados correspondientes a dos niveles de salida.

Circuito integrado LM1524

Este circuito dispone de todos los elementos necesarios para realizar el control de una fuente de alimentación conmutada con regulación por ancho de impulso. Puede utilizarse para generación de tensión de cualquier polaridad con o sin transformadores de salida. Lleva incorporado un generador de tensión continua de baja corriente (hasta 50mA) para poder alimentar los circuitos asociados al de control, con lo que simplifica el diseño completo. Para conseguir el funcionamiento seguro del equipo los transistores de salida están dotados de circuito de protección por limitación de corriente, además de circuito sensor de sobrepasamiento de temperatura, que desactiva las salidas cuando detecta la superación de los límites.

Su esquema de principio es el indicado en la figura 4.

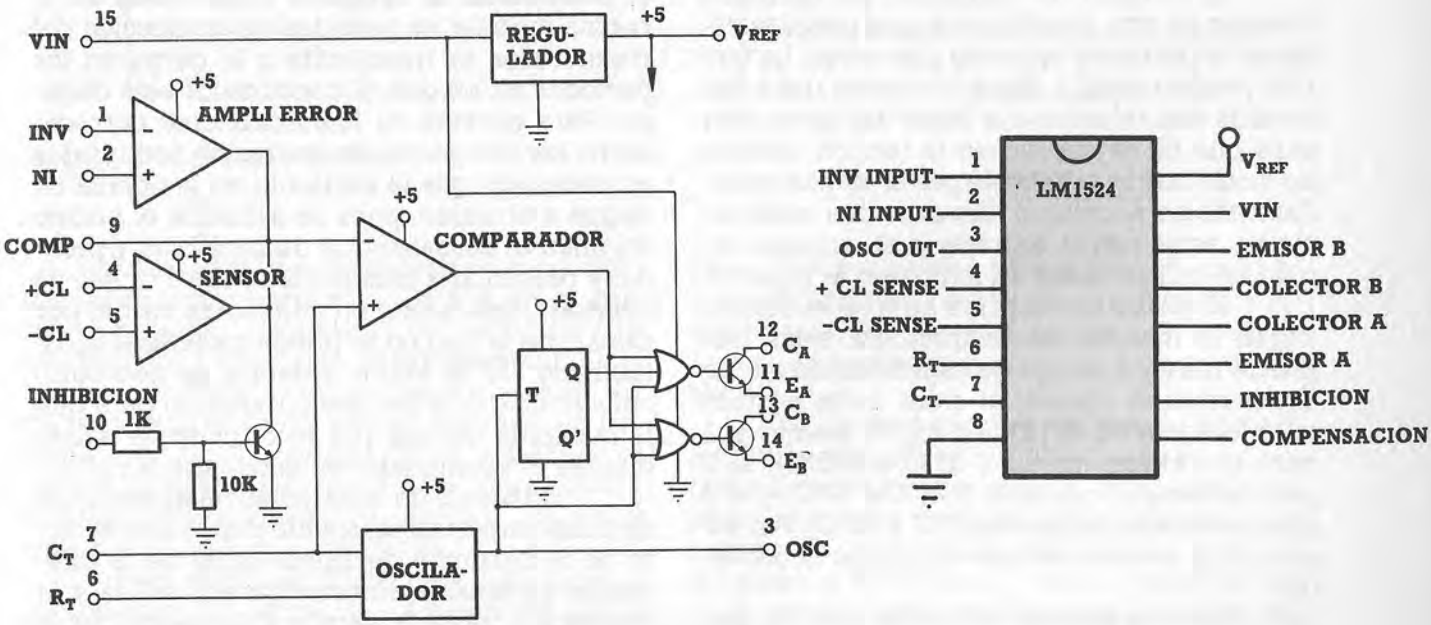


Fig. 4. Diagrama de bloques de LM1524.

El oscilador interno puede hacerse oscilar a cualquier frecuencia dentro del margen 1 Kilohercio a 1 Megahercio, mediante la elección adecuada de la resistencia conectada a la patilla 6 y el condensador conectado a la patilla 7.

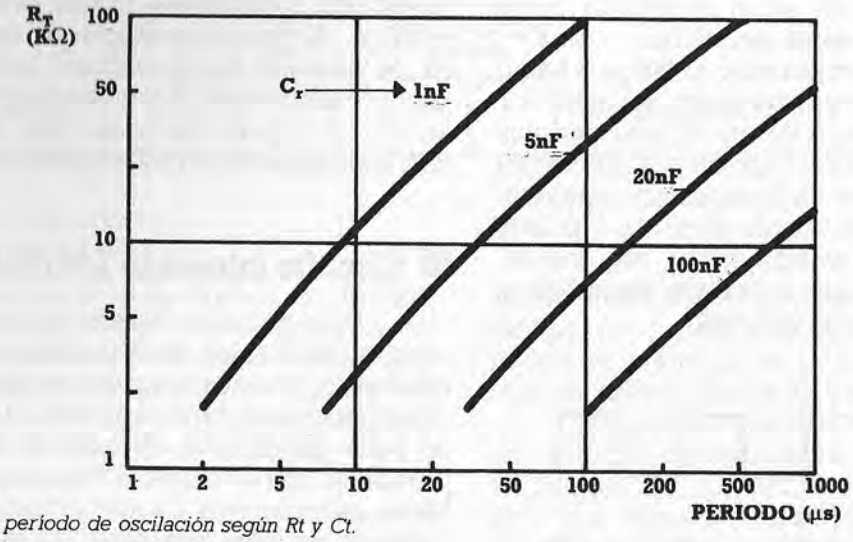


Fig. 5. Gráfico de período de oscilación según Rt y Ct.

El amplificador de error es de tipo diferencial con ganancia muy elevada, pudiendo ajustarse la ganancia en operación mediante los elementos pasivos de realimentación. La salida del amplificador de error está accesible en la patilla 9, para realizar funciones de compensación y protección. La impedancia de salida es alta (> 5 Megaohms), por lo que puede conectarse fácilmente cualquier circuito de inhibición de los pulsos de salida o de control del ciclo útil.

El circuito de limitación de corriente consiste en otro amplificador que permite detectar la corriente aplicada a la carga. La tensión proporcional a dicha corriente debe ser medida con relación a la masa del generador para que no se sobrepase la tensión máxima de modo común admisible por el amplificador. Para ello es necesario disponer una resistencia en serie con la entrada y el terminal común de salida, como se indica en la figura 6.

El circuito se fabrica en tres versiones, según el margen de temperatura a las que puede trabajar dentro de especificaciones. La denominación comercial para cada versión se diferencia en el primer dígito, siendo el 1 para la versión militar (-55°C a 125°C), el 2 para la versión industrial (-25°C a 70°C) y el 3 para la versión comercial (0°C a 70°C). Por supuesto, la tercera versión es la más económica.

Los transistores de salida son de tipo NPN y pueden suministrar hasta un máximo de 100 mA. Se activan en contrafase mediante el circuito divisor por 2 que les precede. Presentan los terminales de emisor y colector independientes para poder ser utilizados en cualquier configuración. Para conseguir corrientes mayores es necesario conectar transistores en acoplamiento Darlington hasta conseguir el valor necesario.

La inductancia utilizada en el circuito es un componente crítico. Como se ha indicado en el principio de funcionamiento de una fuente conmutada, la corriente almacenada en la bobina durante los períodos de conducción del conmutador es transferida a la carga en los períodos en los que el conmutador está cortado. Para permitir el funcionamiento correcto entre los márgenes de operación requeridos es necesario que la corriente en la bobina no llegue a anularse, pues se saturaría el núcleo originando un transitorio de iniciación y parada, y presentaría pulsos a la salida. Por ello es necesario que exista un mínimo de carga, por debajo de la cual no se puede garantizar la estabilidad de la salida. Además es necesario colocar un diodo de libre circulación para que la corriente circule por el circuito de salida cuando el conmutador se encuentra a corte.

El cálculo de la inductancia de salida es de suma importancia, siendo dependiente tanto de la corriente de salida como de la diferencia de tensión entre entrada y salida y la frecuencia de conmutación, de acuerdo con la fórmula:

$L_1 = 2.5\ Vo\ (Vin - Vo) / Io * Vin * F$

El valor se obtiene en Henrios. Para la construcción de estas bobinas se utilizan núcleos de ferrita, que trabajan bien a alta frecuencia.

El condensador de salida se dimensiona principalmente para cumplir las condiciones de rizado necesarias, siendo su valor:

$Co = (Vin - Vo) * Vo * T / 8 * Ri * Vin * L1$

En la ecuación T es el período completo y Ri el rizado máximo requerido. El valor del condensador se obtiene en Faradios.

Circuito práctico

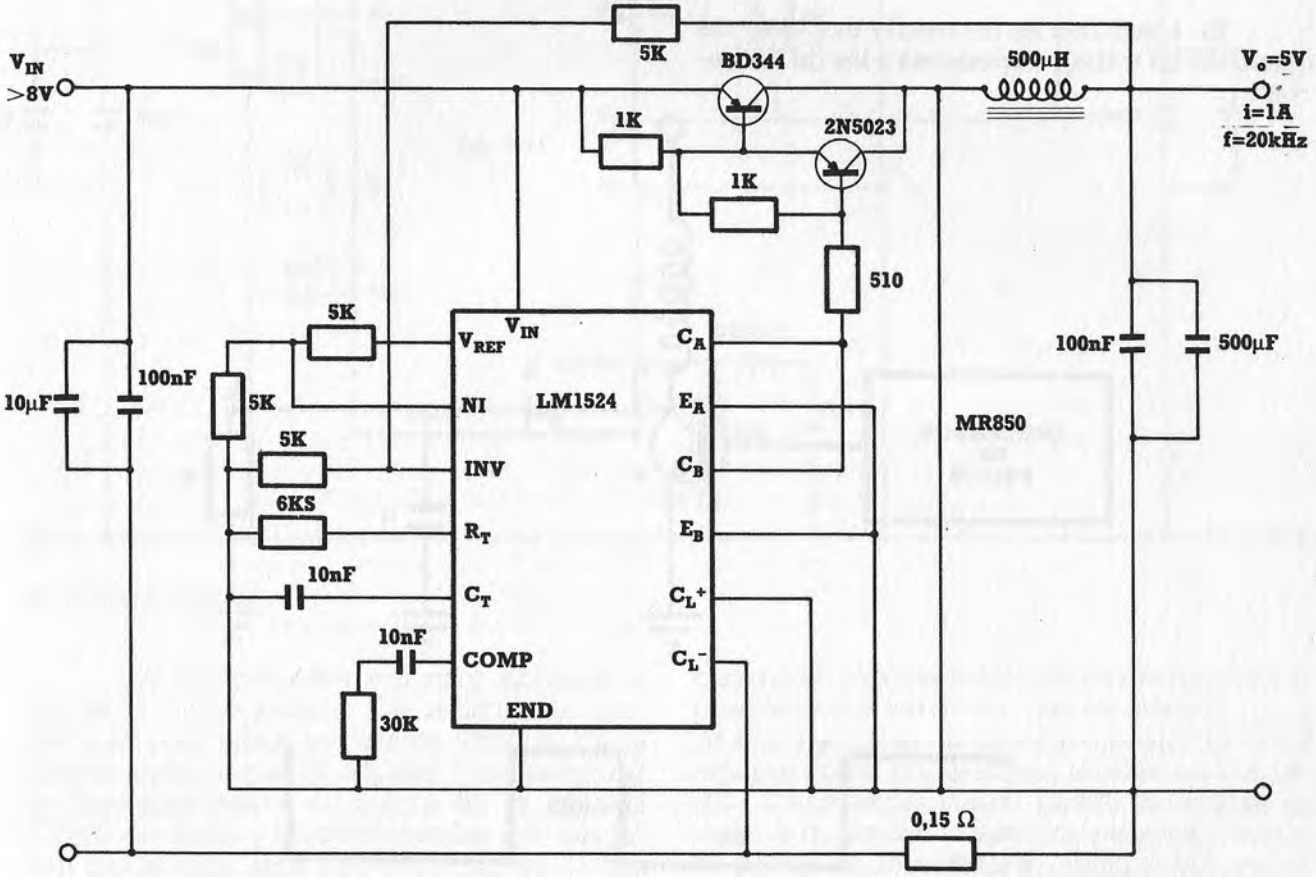


Fig. 6. Fuente conmutada de 5 voltios.

En el esquema se muestra un ejemplo de aplicación del circuito LM1524 en el que para conseguir 1 Amperio de salida se han añadido dos transistores en acoplamiento Darlington. También puede observarse que los transistores de salida se han utilizado en paralelo, con lo que puede llegar a trabajarse con un ciclo de trabajo del 90%, permitiendo trabajar con tensiones de entrada de valor muy poco superior al de salida.

Para tensiones de entrada superiores a 10 voltios y una carga de 1 Amperio se consigue una eficiencia del 80%. El rizado es de unos 10 milivoltios de pico a pico. Se supone una entrada rectificada con un filtrado suficiente para que los mínimos de la entrada no hagan al regulador salirse de su margen de trabajo.

Todos los componentes son de fácil adquisición, escepto la inductancia, que será necesario montarla uno mismo. El núcleo ideal para una bobina de estas características es un toroide de Ferroxcube de 20 milímetros de diámetro y 40 espiras. Puede sustituirse con otros componentes en forma de E, siendo entonces necesario ajustar la inductancia empí-

ricamente. El problema no es sencillo, pues hay que considerar que la bobina trabaja con una corriente continua superpuesta, por lo que el valor medido con medidores de inductancia por resonancia dará un valor muy alejado del real en condiciones de trabajo.

Fuente conmutada elevadora de tensión

En ocasiones es necesario disponer de tensiones de trabajo superiores a las de las baterias de alimentación de los equipos portátiles. Ya se ha mostrado cómo podría realizarse la generación de cualquier tensión con inversores utilizando transformadores. Sin embargo, es posible la obtención de tensiones superiores a las de la alimentación mediante la técnica de conmutación si se invierten las posiciones del transistor y la bobina en el circuito antes explicado.

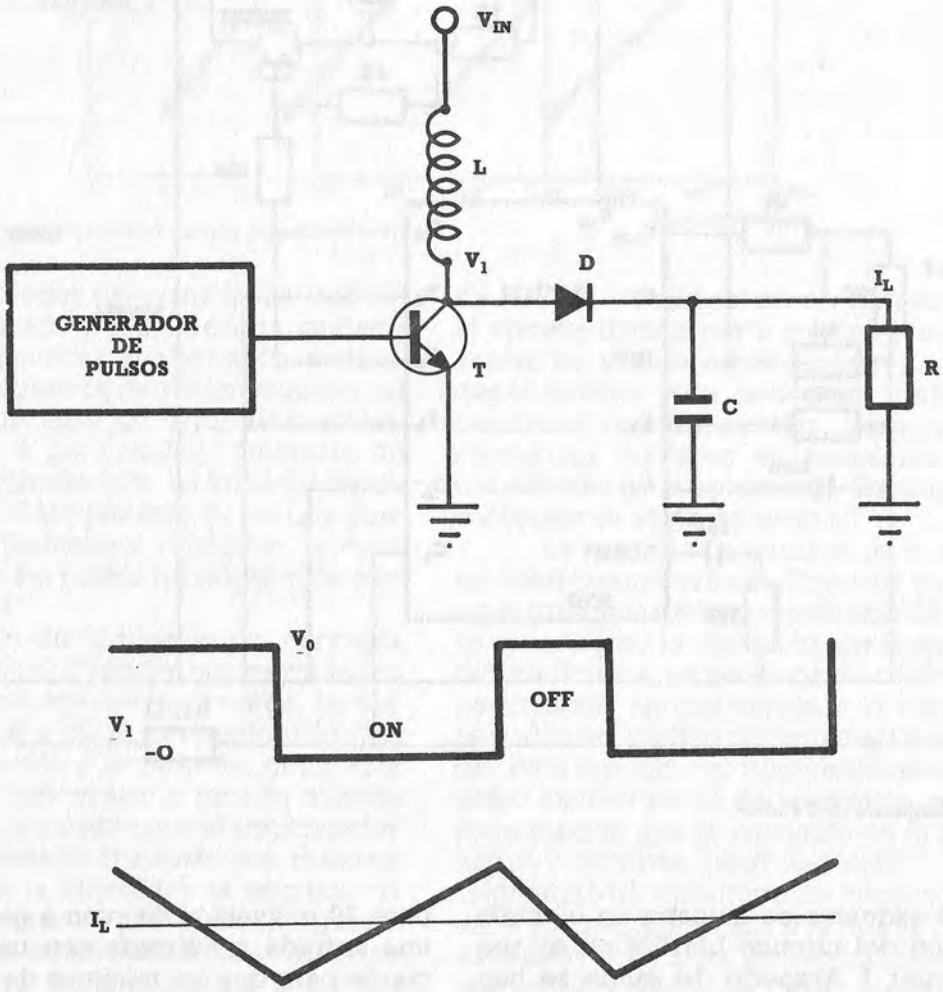


Fig. 7. Fuente conmutada elevadora. Funcionamiento teórico.

El funcionamiento es el contrario que en el caso anterior. Durante el ciclo de conducción del transistor la corriente fluye por la inductacia sin que a la salida se note el cortocircuito, por la presencia del diodo y la bobina. Al abrirse el transistor la energía almacenada en la bobina es transferida a la salida, siendo filtrada su forma por el condensador. El comportamiento transitorio inicial es algo más complejo, siendo cargado el condensador de salida poco a poco por los pulsos aparecidos a la entrada del diodo.

En la práctica el circuito es muy similar al anterior, aunque es necesario indicar que la regulación es menos perfecta para las variaciones de la tensión de alimentación.

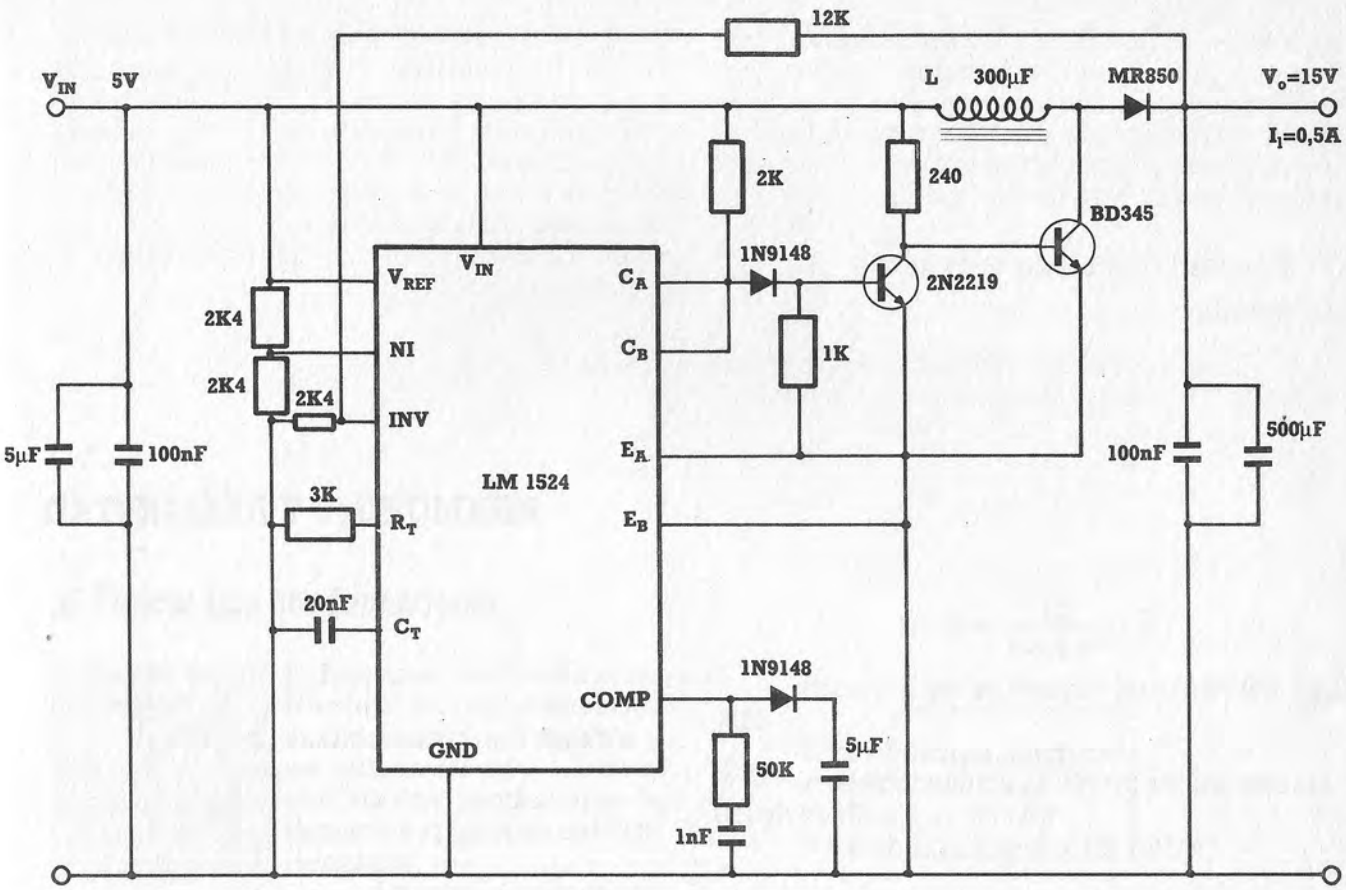


Fig. 8. Circuito práctico.

Los componentes son muy similares a los de la fuente anterior. La inductancia puede montarse sobre un núcleo igual al de la fuente anterior con 24 espiras. Para asegurar un arranque suave se coloca en la entrada COMP un diodo y un condensador que aseguran una subida lenta del ciclo útil, con lo que se garantiza la no saturación del núcleo al arrancar con carga.

Para todas las fuentes conmutadas en general es necesario tomar precauciones sobre la radiación electromagnética originada por la conmutación de fuertes corrientes a alta frecuencia. Es corriente que las fuentes se encierren en carcasa metálica para reducir en lo posible dicha radiación, que en muchos casos podría perturbar el funcionamiento de otros equipos como, por ejemplo, el televisor o la radio. Como experimento puede acercarse la radio a cualquier ordenador personal dotado de fuente de alimentación conmutada y podrá apreciarse en cuántos puntos del dial se «escucha» a la fuente.

A pesar de la menor disipación de potencia de todos los componentes que en el caso de una fuente con regulador lineal es necesario dotar a los transistores con radiadores de disipación.

NATURALEZA Y TECNOLOGIA

Física: Los condensadores

E dice que dos conductores forman un condensador cuando están enfrentados y en influencia total. A cada uno de los conductores le llamamos armadura del condensador.

En un condensador, la relación entre la carga de las armaduras y el potencial entre ambas es una constante que se denomina capacidad del condensador. En el sistema internacional la unidad es el culombio/metro, que se denomina faradio. No obstante, es una unidad demasiado grande, por lo que solemos utilizar múltiplos de ésta. Los más usuales son:

Microfaradio (μF): 10^{-6} faradios.
Nanofaradio (nF): 10^{-9} faradios.
Picofaradio (pF): 10^{-12} faradios.

Cálculo de capacidades de los condensadores más frecuentes

Vamos a realizar el estudio de tres de los condensadores más frecuentes: El esférico, el cilíndrico y el plano.

1. *Condensador esférico*

Si tenemos dos condensadores esféricos enfrentados, de radios R1 y R2 (radio interior y exterior, respectivamente), con cargas iguales -Q y +Q y potenciales V1 y V2, por el teorema de Gauss, en el interior del condensador el campo eléctrico será:

$$E \cdot \vec{a} = \frac{Q}{4\,\pi\,\varepsilon_0\,r^2} \cdot \vec{a}$$

donde *a* es el vector de dirección radial.

E es el campo eléctrico.

r es la distancia al centro de las esferas ($R1 \leq r \leq R2$).

$4\,\pi\,\varepsilon$ es $1/k$, y $k=9 \times 10^9$ (m^2/s^2).

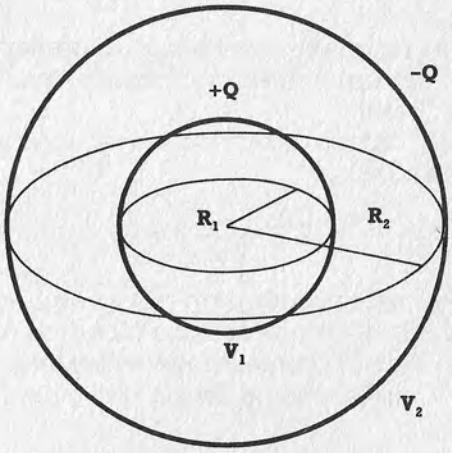


Fig. 1.

La diferencia de potencial entre las dos armaduras vendrá dado por:

$$V1 - V2 = \int_1^2 E \cdot dr = \frac{Q^2}{4 \cdot \pi \cdot \varepsilon o} \cdot \int \frac{dr}{r^2} =$$

$$= \frac{Q}{4 \cdot \pi \cdot \varepsilon o} \cdot \left[\frac{1}{R1} - \frac{1}{R2} \right]$$

de donde sacamos que:

$$C = 4 \cdot \pi \cdot \varepsilon_0 \cdot \frac{R1 - R2}{R1 - R2}$$

Observamos que la capacidad de un condensador no depende del potencial al que está sometido ni de las cargas que contiene, sino únicamente de la geometría del condensador.

2. *Condensador cilíndrico*

El condensador cilíndrico está formado por dos conductores cilíndricos enfrentados (figura 1).

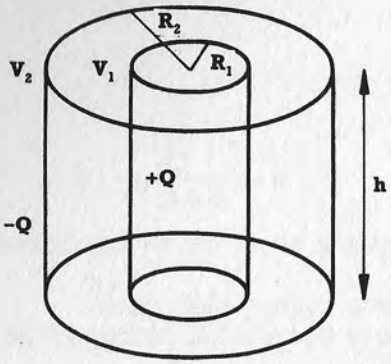


Fig. 2.

El razonamiento teórico es semejante al del condensador esférico, basado en el teorema de Gauss.

El campo eléctrico en el interior del condensador es:

$$E = \frac{Q}{2\pi\varepsilon_0 r h}$$

donde el único parámetro nuevo es *h*, que representa la altura de los dos cilindros (que en nuestro estudio consideramos iguales).

Y de la misma forma que antes llegamos a:

$$C = \frac{Q}{V1 - V2} = \frac{2\pi\varepsilon_0 h}{(*)\ln(R2/R1)}$$

*ln ⟹ LOGARITMO NEPERIANO.

Fijémonos que la capacidad vuelve a depender únicamente de la geometría del diseño.

3. *Condensador plano*

Sean dos conductores plano enfrentados tal como indica la figura 3. Si suponemos el campo uniforme entre las placas y aplicando el teorema de Gauss de nuevo tendremos:

$$E = \frac{\sigma}{\varepsilon 0}$$

donde σ es la densidad superficial de carga.

Aplicando Gauss:

$$V1 - V2 = \frac{\sigma}{\varepsilon 0} \cdot d$$

donde *d* es la distancia entre placas.

Como σ es la densidad superficial de carga ⟹ σ = Q/S luego:

$$V1 - V2 = \frac{Q}{\varepsilon_0 S} \cdot d \Rightarrow C = \varepsilon_0 \cdot \frac{S}{d}$$

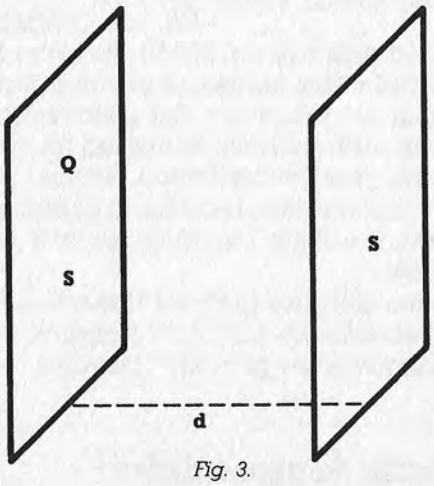


Fig. 3.

```
10 REM*****************************
20 REM*        PROGRAMA           *
30 REM*           DE              *
40 REM*     CONDENSADORES         *
50 REM*****************************
60 REM*    VALIDO PARA AMSTRAD    *
70 REM*    IBM-PC Y COMPATIBLES   *
80 REM*    COMMODORE, SPECTRUM    *
90 REM*          Y  MSX           *
100 REM***************************
105 LET K=9.000001E+09
110 CLS
120 PRINT "ESTE PROGRAMA REALIZA EL CALCULO"
130 PRINT "DE LA CAPACIDAD DE CUALQUIER TIPO"
```

```
140 PRINT "DE CONDENSADORES- ENTRE LOS TRES "
150 PRINT "TRES SIGUIENTES"
160 PRINT :PRINT
170 PRINT "    1.- ESFERICOS"
180 PRINT "    2.- CILINDRICOS"
190 PRINT "    3.- PLANOS"
200 PRINT :PRINT "ELIGE OPCION";
210 INPUT H
220 ON H GOSUB 300,600,900
230 INPUT "DESEAS REALIZAR MAS CALCULOS (S/N) ";W$
240 IF W$="S" THEN GOTO 110
250 END
300 REM***CONDENSADOR ESFERICO
310 PRINT "EL CONDENSADOR ESFERICO ESTA"
320 PRINT "FORMADO POR DOS ESFERA CONCENTRICAS"
330 PRINT "Y LA CAPACIDAD DEPENDE UNICAMENTE"
340 PRINT "DE LOS RADIOS DE LAS ESFERAS."
350 PRINT :PRINT
360 PRINT "INRODUCE RADIO DE LA ESFERA INTERIOR (EN METROS) ";
370 INPUT D
380 PRINT "INTODUCE RADIO DE LA ESFERA EXTERIOR (EN METROS) ";
390 INPUT F
400 IF D>=F THEN PRINT "RADIOS NO VALIDOS":GOTO 350
410 LET C=F*D/((F-D)*K)
415 PRINT
420 PRINT "LA CAPACIDAD DEL CONDENSADOR ES :";C*1000000!;" MICROFARADIOS"
430 PRINT :PRINT
440 RETURN
600 REM ***CONDENSADOR CLILINDRICO
610 PRINT "EL CONDENSADOR CILINDRICO ESTA"
620 PRINT "FORMADO POR DOS CILINDROS CONCENTRICOS"
630 PRINT "Y LA CAPACIDAD DEPENDE UNICAMENTE"
640 PRINT "DE LOS RADIOS DE LOS CILINDROS"
645 PRINT "Y DE LA ALTURA DE ESTOS."
650 PRINT :PRINT
660 PRINT "INRODUCE RADIO DEL CILINDRO INTERIOR (EN METROS) ";
670 INPUT D
680 PRINT "INTODUCE RADIO DEL CILINDRO EXTERIOR (EN METROS) ";
690 INPUT F
700 IF D>=F THEN PRINT "RADIOS NO VALIDOS":GOTO 650
710 PRINT "INTRODUCE ALTURA DE LOS CILINDROS (EN METROS) ";
720 INPUT H
730 LET C=2*H/(K*LOG(F/D))
735 PRINT
740 PRINT "LA CAPACIDAD DEL CONDENSADOR ES :";C*1000000!;" MICROFARADIOS"
750 RETURN
900 REM***CONDENSADOR PLANO
910 PRINT "EL CONDENSADOR PLANO ESTA"
920 PRINT "FORMADO POR DOS PLACAS PARALELAS"
930 PRINT "CON UNA SUPERFICIE -S-, SEPARADAS"
940 PRINT "UNA DISTANCIA -D-."
950 PRINT :PRINT
960 PRINT "INTRODUCE LA SUPERFICIE DE LAS PLACAS (EN METROS) ";
970 INPUT S
980 PRINT "INTRODUCE LA SEPARACION ENTRE PLACAS (EN METROS)_ ";
990 INPUT D
1000 LET C=S/(D*4*3.1415*K)
1005 PRINT
1010 PRINT "LA CAPACIDAD DEL CONDENSADOR ES :";C*1000000!;" MICROFARADIOS"
1020 RETURN
2000 REM*********************************
2010 REM* MODIFICACIONES PARA SPECTRUM *
2020 REM*  220 LET P=300*H              *
2030 REM*  225 GOSUB P                 *
2040 REM*********************************
2050 REM
2060 REM
3000 REM**********************************
3010 REM* MODIFICACIONES PARA COMMODORE *
3020 REM* 110 PRINT CHR$(115)           *
3030 REM**********************************
```

MATEMATICAS

Cónicas

Como continuación al programa del otro día, vamos a ver hoy un programa de reconocimiento de cónicas a través de su ecuación en forma matricial. NOTA MUY IMPORTANTE: recordemos que las matrices de una cónica tienen que ser SIMETRICAS.

Este programa analiza una cónica introducida a través de una matriz y determina el

tipo de cónica y su ecuación (en cartesianas y homogéneas).

Recordemos que la ecuación de una cónica en cartesianas viene dada por la expresión:

$$a_{11}x^2 + 2a_{12}xy + 2a_{13}X + a_{22}y^2 + 2a_{23}y + a_{33} = 0$$

Por tanto, la cónica queda definida como el lugar geométrico de los puntos del plano cartesiano que cumplen dicha ecuación.

Pero a veces se nos presentan determinados puntos que no pueden ser estudiados en el plano cartesiano. Son los llamados puntos impropios. Así, mientras que en el plano cartesiano somos incapaces de representar el punto de cruce de dos rectas paralelas (el punto del infinito), en coordenadas homogéneas ese punto se representa como (a,b,0), donde (a,b) es el vector director de las dos rectas. Por tanto, llegamos a la conclusión de que si la última cifra de las coordenadas homogéneas es cero, entonces es un punto impropio. Si esta tercera cifra es distinta de cero, entonces es un punto real.

De acuerdo con esto, la ecuación en coordenadas homogéneas será:

$$a_{11}x^2 + 2a_{12}xy + 2a_{13}xt + a_{22}y^2 + 2a_{23}yt + a_{33}t^2 = 0$$

Cónicas: Clasificación

Las cónicas se clasifican según dos formas:

— Su determinante.

— El valor del elemento a_{33}.

Según la primera clasificación, las podemos diferenciar en dos grupos:

a) Reguladores (DET(A)<>0).
b) Degeneradas (DET(A) = 0).

Según la segunda se pueden clasificar en:

a) Elipses.
b) Hipérbolas.
c) Parábolas.

Y dentro de cada grupo se subdividen en más subgrupos:

a) *Elipses*
a.1) Si DET(A)<>0→ ELIPSE
a.1.1) Si DET(A)*(a_{11}+a_{12})>0 IMAGINARIA
a.1.2) Si DET(A)*(a_{11}+a_{12})<0 REAL
a.2) Si DET(A) = 0 → 2 RECTAS IMAGINARIAS QUE SE CORTAN EN UN PUNTO SINGULAR.

b) *Hipérbolas*
b.1) Si DET(A)<>0 HIPERBOLA.
b.2) Si DET(A) = 0 DOS RECTAS REALES QUE SE CORTAN EN UN PTO. SINGULAR.

c) *Parábolas*
c.1) Si DET(A)<> PARABOLA.
c.2) Si DET(A) = 0
c.2.1) Si a_{11}=0 y a_{22}=0 RECTA DOBLE.
c.2.2) Si a_{11}>0 o a_{22}>0 2 RECTAS PARALELAS IMAGINARIAS.
c.2.3) Si a_{11}<0 o a_{22}<0 2 RECTAS PARALELAS REALES.

```
10 REM*****************************
20 REM*        PROGRAMA           *
30 REM*           DE              *
40 REM*        CONICAS            *
50 REM*****************************
60 REM*   VALIDO PARA AMSTRAD     *
70 REM*   IBM-PC Y COMPATIBLES    *
80 REM*   COMMODORE, SPECTRUM     *
90 REM*          Y  MSX           *
100 REM****************************
110 CLS
120 PRINT "EN ESTE PROGRAMA INTRODUCIMOS"
130 PRINT "UNA CONICA Y POSTERIORMENTE"
140 PRINT "RECONOCE EL TIPO DE CONICA QUE ES"
150 PRINT :PRINT
160 PRINT "INTRODUCE LA CONICA EN FORMA MATRICIAL"
170 FOR J=1 TO 3
180 FOR I=1 TO 3
190 PRINT "INTRODUCE ELEMENTO ";I;",";J;
200 INPUT A
210 LET C(I,J)=A
220 NEXT I
230 NEXT J
240 PRINT
270 FOR J=1 TO 3
280 FOR I=1 TO 3
290 IF C(I,J)<>C(J,I) THEN PRINT "ESTO NO ES UNA CONICA":GOTO 160
```

```
320 NEXT I
330 NEXT J
344 PRINT "LA ECUACION EN CARTESIANAS ES : "
345 PRINT "(";C(1,1);")X²+ 2(";C(1,2);")XY +2(";C(1,3);")X +(";C(2,2);")Y²+2(";C
(2,3);")Y + ";C(3,3)
346 PRINT "LA ECUACION EN HOMOGENEAS ES  : "
347 PRINT " (";C(1,1);")X²+ 2(";C(1,2);")XY +2(";C(1,3);")XT +(";C(2,2);")Y²+2("
;C(2,3);")YT +(";C(3,3);")T²"
348 PRINT :PRINT
349 GOSUB 1000
350 IF DET=0 THEN PRINT " => ES UNA CONICA DEGENERADA":GOSUB 2000
360 IF DET<>0 THEN PRINT " => ES UNA CONICA REGULAR":GOSUB 3000
999 END
1000 REM***DETERMINANTE 3X3*****
1010 LET DET=C(1,1)*(C(2,2)*C(3,3)-C(2,3)^2)-C(1,2)*(C(1,2)*C(3,3)-C(2,3)*C(1,3)
)+C(1,3)*(C(1,2)*C(2,3)-C(2,2)*C(1,3))
1020 PRINT "EL DETERMINANTE VALE";DET
1025 PRINT :PRINT
1030 RETURN
2000 REM***DEGENERADAS****
2010 IF C(3,3)<0 THEN PRINT "LA CONICA DEGENERADA SON DOS RECTAS REALES QUE SE
CORTAN EN UN PUNTO SINGULAR"
2020 IF C(3,3)>0 THEN PRINT "LA CONICA DEGENERADA SON DOS RECTAS IMAGINARIAS QUE
 SE CORTAN EN UN PUNTO REAL"
2030 IF C(3,3)<>0 THEN GOTO 2500
2040 IF (C(1,1)=0) AND (C(2,2)=0) THEN PRINT "LA CONICA DEGENERA ES UNA RECTA RE
AL DOBLE."
2050 IF (C(1,1)>0) OR (C(2,2)>0) THEN PRINT "LA CONICA DEGENERADA SON DOS RECTAS
 PARALELAS IMAGINARIAS
2060 IF (C(1,1)<0) OR (C(2,2)<0) THEN PRINT "LA CONICA DEGENERADA SON DOS RECTAS
  REALES."
2500 RETURN
3000 REM***REGULARES  ****
3010 IF C(3,3)=0 THEN PRINT "ES UNA CONICA REGULAR TIPO PARABOLICO":GOTO 3050
3020 IF C(3,3)<0 THEN PRINT "ES UNA CONICA REGULAR TIPO HIPERBOLICO":GOTO 3050
3025 LET H=((C(1,1)+C(1,2))*DET)
3030 IF H>0 THEN PRINT "ES UNA CONICA REGULAR TIPO ELIPTICO  IMAGINARIO"
3040 IF H<0 THEN PRINT "ES UNA CONICA REGULAR TIPO ELIPTICO REAL. "
3050 RETURN
4000 REM ***********************************
4010 REM * MODIFICACIONES PARA COMMODORE   *
4020 REM * CAMBIAR LOS CLS POR :           *
4030 REM *  PRINT CHR$(115)                *
4040 REM ***********************************
```

SOCIEDAD

Informática: Un programa básico monitor

En este capítulo vamos a tratar directamente el tema de la informática. Muchas veces habrá oído hablar de los ordenadores, y muchas veces habrá pensado en los misterios de la informática. En este capítulo vamos a intentar que ese mundo lejano no parezca tan incomprensible como usted cree.

Un ordenador lo que básicamente hace es tratar datos. Ahora bien, esos datos no están representados tal y como nosotros los representamos en la vida cotidiana. Esto es debido a que un ordenador es incapaz de entender el código usado por nosotros. Igual que nosotros usamos un código en el que distinguimos una palabra de otra por la entonación, tono..., el ordenador sólo es capaz de detectar dos tipos de tonos: existencia de corriente eléctrica y no existencia de corriente eléctrica. Este es el llamado código binario. En capítulos anteriores vimos que existen distintas bases numéricas (decimal, octal, binaria...), pues bien, el ordenador trata toda la información en base 2 (o sea, binario); así, por ejemplo, la letra "A" será 10010010 y la palabra "PEPE" vendrá representada para el ordenador como, por ejemplo:

10010011 00001001 10010011 00001001

Por supuesto que nosotros, cuando programamos o ejecutamos un programa, no empleamos este código, ya que el ordenador uti-

liza algún "traductor" que nos hace más fácil nuestra tarea. No obstante, resulta interesante ver cómo almacenamos realmente la información en el ordenador.

Cada dígito (1 ó 0) es lo que llamamos un BIT, que es la unidad básica de información. El ordenador agrupa los bits en grupos llamados PALABRA. En nuestros ordenadores generalmente la palabra consta de 8 BITS, que usualmente llamamos BYTE. Así, cuando nosotros decimos que nuestro ordenador tiene 48 K, 64 K, 16 K..., a lo que nos referimos es a que tiene (por semejanza con la magnitud KILO = 1000) *aproximadamente* 48.000 BYTES o 64.000 BYTES... Decimos aproximadamente porque 1 Kbyte = 1.024 bytes. Es debido a que el KILOBYTE = 2^{10} bytes.

PEEK y POKE

Muchas veces al hojear el manual de BASIC hemos encontrado que existen dos sentencias que nunca hemos descubierto muy bien qué hacen. Nos referimos a PEEK y a POKE. Vamos a realizar un pequeño estudio de ambas, pues de ellas depende el programa desarrollado en este apartado. Antes de nada conviene poner una cosa clara: igual que la sentencia PEEK (husmear) no tiene ningún tipo de riesgo usarla, la sentencia POKE (hurgar) puede echar por tierra horas de trabajo si no se utiliza bien, así que recomendamos su uso cuando no tengamos nada importante en memoria o cuando ya se conozca lo suficiente para utilizarla sin peligro.

PEEK (dirección de memoria). Esta sentencia lo que hace es hallar el contenido de la dirección de memoria especificada entre paréntesis. El resultado es un número comprendido entre 0 y 255 (recuérdese que la longitud del BYTE = 8 bits ⇒ el número estará entre 00000000 = 0 en decimal y 11111111 = 255 en decimal) y podemos realizar cuantos PEEKs queramos sin tocar para nada su contenido.

POKE (dirección de memoria, nuevo contenido). Esta sentencia realiza la siguiente función. Escribe (y, por tanto, borra el contenido anterior) en la dirección de memoria especificada, un nuevo número (nuevo contenido) comprendido entre 0 y 255. El peligro de esta sentencia es que si no dominamos bien el ordenador, podemos escribir en zonas que ocupa el sistema o nuestro programa, por lo que borraremos el contenido anterior. No obstante, es importante que practiques con los POKEs, por lo que, si es tu primer contacto con el lenguaje máquina, lo mejor que puedes hacer es no tener en memoria nada importante (o, por ejemplo, el programa monitor), y si llegas a un punto donde falla el ordenador, lo desenchufes y lo vuelvas a enchufar, con lo que se reinicializa todo el sistema (hacer un RESET).

Numeración hexadecimal. Hemos visto que el sistema binario es el utilizado por el ordenador para su funcionamiento. Sin embargo, programar en código binario es, aparte de más o menos difícil, muy largo. Pensemos, por ejemplo, que el número 255 en base 10 lo representamos como 11111111.

Surge entonces un problema: necesitamos programar en código máquina pero el sistema binario es demasiado pesado. La primera solución que se nos ocurre es programar directamente en decimal. El resultado es que, aunque más fácil, es mucho más tedioso, pues necesitamos siempre convertir números binarios a decimales y viceversa, y además, el contenido de cada bit que se veía a simple vista en el código binario es totalmente indescifrable en decimal. Conclusión, necesitamos un código más manejable que el binario, pero en el que el contenido de cada bit sea rápidamente identificable. Un sistema de numeración que lo cumple perfectamente es el HEXADECIMAL (numeración en base 16 = 2^4), ya que, por un lado, su representación es más corta (incluso más que el decimal), y por otro lado, si bien no inmediatamente, pero con gran facilidad, podemos adivinar el contenido de cada bit.

Binario	Decimal	Hexadecimal
0000 0000	0	00
0000 0001	1	01
0000 0010	2	02
0000 0011	3	03
0000 0100	4	04
0000 0101	5	05
0000 0110	6	06
0000 0111	7	07
0000 1000	8	08
0000 1001	9	09
0000 1010	10	0A
0000 1011	11	0B
0000 1100	12	0C
0000 1101	13	0D
0000 1110	14	0E
0000 1111	15	0F
0001 0000	16	10

No hay más que aprenderse la notación de los 15 primeros números hexadecimales, y su conversión binaria es casi inmediata:

Por ejemplo: El número F5 en hexadecimal:

— sabemos que F = 15 = 1111

— sabemos que 5 = 5 = 0101, luego el número en binario será: 1111 0101

```
10 REM*************************
20 REM*        PROGRAMA        *
30 REM*           DE__         *
40 REM* MONITOR DE COD. MAQ    *
50 REM*************************
60 REM*   VALIDO PARA AMSTRAD  *
70 REM*   IBM-PC Y COMPATIBLES *
80 REM*   COMMODORE, SPECTRUM  *
90 REM*        Y  MSX          *
100 REM*************************
105 CLS
110 PRINT "ESTE PROGRAMA TE PERMITE LEER"
120 PRINT "O MODIFICAR POSICIONES DE "
130 PRINT "MEMORIA EN NOTACION HEXADECIMAL"
140 PRINT
150 PRINT "            MENU"
160 PRINT "            ===="
170 PRINT :PRINT
180 PRINT " 1.- LEER DIRECCIONES Y DATOS"
190 PRINT " 2.- MODIFICAR CONTENIDO DE POSICIONES"
200 PRINT :PRINT
210 INPUT "INTRODUCE OPCION";H
220 ON H GOSUB 600,900
230 GOTO 10
600 REM**** LEER POSICIONES****
605 CLS
610 INPUT "INTRODUCE DIRECCION DE COMIENZO DE LECTURA (EN HEX) ";HX$
620 GOSUB 2000
640 PRINT "    DIRECCION",,"    CONTENIDO"
645 PRINT "DEC","HEX","DEC","HEX"
646 PRINT "===","===","===","==="
650 FOR J=DEC TO DEC+16
660 LET N=J
670 GOSUB 1000
680 PRINT J,P$,
690 LET N=PEEK(J)
700 GOSUB 1000
705 PRINT PEEK(J),P$
710 NEXT J
720 INPUT "PULSA ENTER PARA CONTINUAR";W$
730 RETURN
900 REM****MODIFICAR POSICIONES*****
910 CLS
920 INPUT "QUE POSICION DESEAS MODIFICAR (EN HEX.) ";HX$
921 LET PO$=HX$
925 GOSUB 2000
930 LET PS=DEC
935 INPUT "QUE NUEVO VALOR QUIERES ASIGNARLE (EN HEX.) ";HX$
936 LET JX$=HX$
937 GOSUB 2000
940 LET CON=DEC
941 PRINT PS,CON
945 POKE PS,CON
950 PRINT "HAS CAMBIADO EL CONTENIDO DE ";PO$;" POR EL VALOR ";JX$
960 INPUT "DESEAS UNA NUEVA MODIFICACION (S/N) ";W$
965 IF W$="S" THEN GOTO 920
970 RETURN
1000 LET P$=""
1005 R=((N/16)-INT(N/16))*16
1010 GOSUB 1070
1020 LET N=INT(N/16)
1030 IF N=>16 THEN GOTO 1005
1040 LET R=N
1050 GOSUB 1070
1060 RETURN
1070 IF R>9 THEN GOTO 1110
1080 LET ASI=ASC("0")
1090 LET RES=0
1100 GOTO 1130
1110 LET ASI=ASC("A")
1120 LET RES=10
1130 LET H=R+ASI-RES
```

```
1140 LET C$=CHR$(H)
1150 LET P$=C$+P$
1160 RETURN
2000 LET L=LEN (HX$)
2005 LET DEC=0
2010 FOR J=1 TO L
2020 LET D$=LEFT$(HX$,1)
2030 GOSUB 2070
2040 LET DEC=DEC+C
2050 NEXT J
2060 RETURN
2070 LET P=ASC(D$)-ASC("A")
2080 IF P=>0 THEN C=P+10:GOTO 2100
2090 LET C=VAL(D$)
2100 LET HX$=RIGHT$(HX$,L-J)
2110 LET C=C*16^(L-J)
2120 RETURN
3000 REM ***********************************
3010 REM * MODIFICACIONES PARA COMMODORE *
3020 REM *  CAMBIAR DONDE PONGA CLS POR: *
3030 REM *      PRINT CHR$(115)          *
3040 REM ***********************************
3050 REM ***********************************
3060 REM ***********************************
3070 REM *  MODIFICACIONES PARA SPECTRUM *
3080 REM * CAMBIAR DONDE PONGA -ASC- POR *
3090 REM *          CODE                 *
3091 REM * LINEA 220 LET H=(H+1)*300     *
3092 REM * LINEA 221 GOSUB H             *
3100 REM ***********************************
```

PARA LOS MAS JOVENES

Husos horarios

Los husos horarios son cada una de las 24 partes de 15.° en que está dividida la superficie terrestre, mediante sus meridianos (círculo máximo de la esfera terrestre que pasa por los polos). La Tierra se ha dividido de polo a polo en 360 medias circunferencias o meridianos: 180 al Este y 180 al Oeste. Se toma como referencia el meridiano 0°, que pasa por Greenwich (viejo observatorio próximo a Londres). El meridiano opuesto, el 180°, atraviesa el océano Pacífico.

El día solar se divide en períodos de tiempo a los que llamamos horas. En el movimiento de rotación todos los puntos de un mismo meridiano pasan frente al Sol al mismo tiempo y, por tanto, todos tienen la misma hora solar. Pero como todos los meridianos pasan sucesivamente frente al Sol, tendría que haber 360 horas distintas en la Tierra. Esto tendría grandes inconvenientes, por lo que se ha dividido la esfera terrestre en 24 agrupaciones de 15 meridianos, llamados husos horarios. Toda la superficie comprendida en esos 15 meridianos tiene la misma hora.

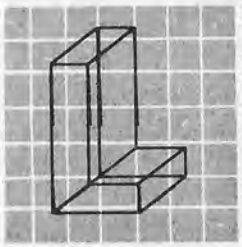
Los husos horarios no siguen siempre la trayectoria de los meridianos, sino que procuran adaptarse a las fronteras de los países, para que en el interior de los países rija siempre la misma hora legal. Esto no puede suceder en países muy extensos como, por ejemplo, Canadá y la Unión Soviética, en los que existen varias horas legales.

Los viajeros que se dirigen hacia el Oriente tiene que adelantar sus relojes cada 15° de meridiano; y los viajeros que se dirigen hacia el lado contrario, deben retrasar una hora. En el meridiano 180° los que vienen de Occidente retrasan un día su calendario, y los que proceden de Oriente adelantan un día su calendario. Por tanto, los husos horarios significan la medición internacional de la hora.

```
10 REM*****************************
20 REM*        PROGRAMA           *
30 REM*           DE              *
40 REM*     HUSOS HORARIOS        *
50 REM*****************************
60 REM*   VALIDO PARA AMSTRAD     *
70 REM*   IBM-PC Y COMPATIBLES    *
80 REM*   COMMODORE, SPECTRUM     *
90 REM*          Y  MSX           *
100 REM****************************
```

```
105 RESTORE
110 CLS
120 PRINT "ESTE PROGRAMA LE CALCULA LA HORA"
130 PRINT "DE CUALQUIER CIUDAD DEL MUNDO"
140 PRINT "CUANDO ESTAMOS EN OTRA PARTE"
150 PRINT "DEL MUNDO."
160 PRINT
170 INPUT "PULSA ENTER PARA CONTINUAR";W$
180 CLS
190 PRINT :PRINT
200 INPUT "INTRODUCE CIUDAD EN LA QUE ES ENCUENTRA ";C$(1)
210 LET K=1
220 GOSUB 1000
225 RESTORE
230 INPUT "INTRODUCE CIUDAD EN LA QUE SE QUIERE SABER LA HORA";C$(2)
240 LET K=2
250 GOSUB 1000
260 PRINT "INTRODUCE LA HORA (HH.MM) EN ";C$(1)
270 INPUT H1
271 GOSUB 2000
274 LET T$=""
275 IF INT(H1)=H1 THEN LET T$=".00"
280 LET DIF=C(2)-C(1):LET D$=" MISMO DIA"
290 LET H2=H1+DIF
291 IF H2<=0 THEN LET H2=H2+24:LET D$=" DIA ANTERIOR"
292 IF H2>24 THEN LET H2=H2-24:LET D$=" DIA SIGUIENTE"
300 PRINT"CUANDO SON LAS ";H1;T$;" EN ";C$(1);" SON LAS ";H2;T$;" EN ";C$(2);" D
EL":D$
310 END
1000 REM SUBRUTINA
1010 LET J=1
1015 LET F=0
1020 READ A$
1030 IF A$=C$(K) THEN F=1
1040 READ A
1050 IF F=1 THEN C(K)=A
1060 LET J=J+1
1065 IF J<=50 AND F=0 THEN GOTO 1015
1066 IF F=0 THEN INPUT "ESA CIUDAD NO LA CONOZCO, PULSA ENTER PARA EMPEZAR ";W$:
GOTO 10
1070 RETURN
2000 REM****COMPROBACION DE HORA*****
2010 IF H1>24 OR H1<=0 THEN PRINT "HORA INCORRECTA":GOTO 260
2020 RETURN
3000 DATA MADRID,0,MEXICO,-6,LISBOA,0,PARIS,0,LONDRES,0,ROMA,1,BERLIN,1,OSLO
3010 DATA 1,ESTOCOLMO,1,ESTAMBUL,2,MOSCU,2,TOKIO,9,SHANGAI,8,SYDNEY,10
3020 DATA WELLINGTON,12,FORMOSA,8,EL CABO,1,PRETORIA,2,LAGOS,1,DAKAR,0,RABAT
3040 DATA 0,TRIPOLI,1,EL CAIRO,2,TEHERAN,3,DAR EE SAALAM,3,REYKJAVIK,-1,CANARIAS
3050 DATA -1,CABO VERDE,-2,BOGOTA,-5,LIMA,-5,LA PAZ,-4,SANTIAGO,-5,OMSK,+5
3060 DATA MALVINAS,-4,BUENOS AIRES,-4,MONTEVIDEO,-3,RIO DE JANEIRO,-3,SALVADOR
3070 DATA -3,BELEM,-3,CARACAS,-4,PEKIN,8,LA HABANA,-5,SAN FRANCISCO,-8,LASA,6
3080 DATA DENVER,-7,CHICAGO,-6,OTTAWA,-5,NEW YORK,-5,HAWAI,-10,NUEVA DELHI,5
4000 REM ********************************
4010 REM * MODIFICACIONES PARA COMMODORE*
4020 REM * DONDE PONGA  CLS CAMBIAR POR *
4030 REM *        PRINT CHR$(115)        *
4040 REM ********************************
```

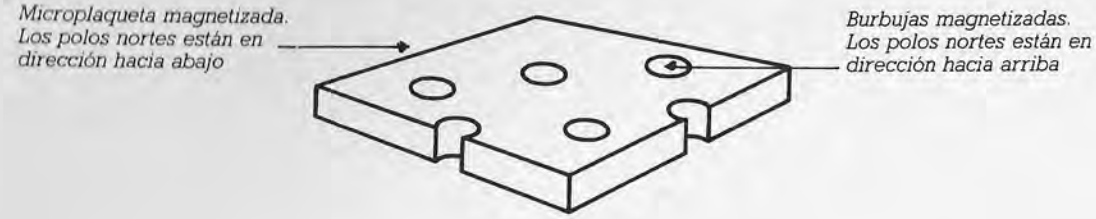
Cuarta generación de ordenadores



O que sí es claro es que pueden distinguirse fácilmente tres generaciones de ordenadores, ya que existen factores diferenciadores muy evidentes. Los ordenadores de la primera generación eran enormes, pesados, frágiles. La segunda generación de ordenadores se distinguió por la aparición del transistor, que hizo mucho menores, menos frágiles y más fiables a las atractivas y esotéricas máquinas, a las que muchos llamaban "cerebros electrónicos". La tercera generación fue mucho más allá. No se contentó con la disminución de tamaño de los ordenadores. Fue un intento serio de eliminar conductores, reducir tiempos, simplificar reparaciones, eliminar soldaduras, en suma, simplificar al máximo el hardware de las máquinas con la aparición de los circuitos integrados. La simplificación del hardware de la máquina no fue, sin embargo, la única aportación de esta tercera generación. Las mejoras de una a otra generación han sido muy importantes. Los ordenadores de la tercera generación ya pasan a estar al alcance de las empresas, el hombre de la calle ya comienza a interesarse en la labor que realizan, en suma, el ordenador pasa a ser un objeto (si es que se le puede llamar objeto) cotidiano. ¿Cómo definiríamos esta tercera generación? Sin duda, por la aparición del circuito integrado. No pensemos, sin embargo, que los circuitos integrados son ya elementos del pasado. El circuito integrado ha sido el punto de partida de un camino que en nuestros días sigue teniendo vigencia y sentido. Naturalmente, los grados de integración de la tercera generación frente a los posibles en nuestros días pueden provocar sonrisas, pero han servido para hacer evidente un campo, hasta aquel momento sin explorar, del que se podía sacar muchísimo provecho.

Respecto de la cuarta generación, el elemento que la caracteriza es la aparición del microprocesador. Un microprocesador es, en palabras sencillas de la calle, un pequeño ordenador contenido en una pastilla. Las ventajas de este pequeño invento son enormes. Y, entre ellas, es muy importante el precio.

El nacimiento del microprocesador data de 1969, cuando una compañía japonesa, Busicom, encarga a la Intel Corporation una serie de microplaquetas de circuitos integrados para sus calculadoras. La Intel era una compañía que llevaba poco tiempo trabajando. Había sido fundada año y medio antes por Robert Noyce, que conocía perfectamente el mercado de ordenadores, circuitos integrados, microplaquetas, etc. El proyecto de diseño de estas microplaquetas fue asignado a Ted Hoff, que fue el verdadero "padre" del microprocesador. Ted Hoff decidió simplificar el problema, y en lugar de diseñar once tipos distintos de microplaquetas, decidió diseñar una única plaqueta, programable para que realizara las labores de las once plaquetas previstas. Esta microplaqueta universal no sólo podía programarse de estas once formas, sino que realmente constituía un pequeño procesador, capaz de ser programado como cualquier otro ordenador. La plaqueta fue llamada Intel 4004, y era una verdadera maravilla. En una superficie de unos 4,5 × 3,5 centímetros se encontraba un circuito integrado compuesto por más de dos mil transistores. La 4004 era un verdadero procesador central, cumpliendo las mismas tareas, aunque con muchísimas más ventajas. Imaginemos cuáles podían ser esas ventajas.



Las burbujas magnéticas

En primer lugar, la plaqueta estaba formada por un único circuito integrado, lo que disminuía enormemente los riesgos de averías. Es evidente que a menor número de componentes, menor número de averías. Sin embargo, ¿tenía las mismas características que un procesador central convencional? Prácticamente las mismas, aunque era más lenta. Pero era mucho más fiable.

Dos años más tarde Intel sacó al mercado un microordenador elemental, el MCS-4, que tenía un precio muy bajo (200 dólares), aunque realmente no había sido pensado como tal ordenador universal, sino como máquina de una sola tarea. El MCS-4 se vendió muchísimo para todo tipo de aplicaciones, como, por ejemplo, el control de las luces de tráfico, control de ascensores, etc. Tenía una ventaja primordial, era una microplaqueta que sustituía a todo un grupo de microplaquetas, y, además, si se deseaba cambiar la función, bastaba con programarla, eliminando conexiones y errores. Su aparición en el año 1971 supuso el comienzo de la caída de precios de los ordenadores hasta las cotas que tenemos en fechas actuales. Estos pequeños microprocesadores aparecen por todas partes, desde máquinas tragaperras, expendedoras de tabacos, etcétera, hasta satélites.

Aunque el microprocesador es el signo distintivo de la cuarta generación de ordenadores, no es el único avance de importancia acaecido en esas fechas. El salto tecnológico ha sido importante. La miniaturización siguió su curso, pero el software y las comunicaciones también mejoraron sustancialmente.

En el año 1969 Andrew Bobeck diseña la memoria magnética de burbujas. Poco tiempo estuvo probándose esa memoria en los Laboratorios de Desarrollo de la Bell, y ese mismo año salió al mercado. Su vigencia duró una década aproximadamente. Entre sus ventajas, están el bajo precio, y, sobre todo, que los datos se podían almacenar largos períodos de tiempo.

En ese mismo año, 1969, aparece otro dispositivo, diseñado también en los Laboratorios de la Bell. El dispositivo CCD (dispositivo con acoplamiento de cargas). Fue diseñado por Elwood Smith y Bill Boyle. Se trata de una memoria en la que las cargas se almacenan bajo los electrodos. Para pasar de un electrodo a otro, basta aplicar el voltaje adecuado. Estos voltajes están dentro de una gama casi continua.

Tres años más tarde, en 1972, IBM lanza al mercado un pequeño dispositivo de una practicidad extraordinaria: el disco magnético flexible. Hay que tener en cuenta que los discos duros resultaban muy caros, por lo que el éxito de estos discos estaba prácticamente asegurado. Además de esta ventaja, el disco flexible presentaba muchas más, entre otras su portabilidad, y un sistema de garantizar la privacidad sin ninguna complicación. Los discos flexibles *(floppys)* tuvieron enorme éxito y difusión, sobre todo para ordenadores de bajo precio. Su calidad era suficiente, y, aunque los primeros no podían almacenar mucha información (unas quince páginas de libro aproximadamente), poco a poco han ido perfeccionándose aumentando tanto la calidad de las grabaciones como su capacidad, que se ha multiplicado por ocho.

En los años que siguieron a la aparición del primer microprocesador, muchas compañías se fueron interesando por él. La propia Intel sacó otro modelo de microprocesador más poderoso que el 4004, el 8008. Sin embargo, como procesador seguía siendo bastante lento. Tuvo un enorme éxito comercial, lo que sirvió de acicate para las otras empresas, que estaban a la expectativa. En el año 1974 apareció el modelo 6800 de Motorola, de 8 bits, que tuvo un enorme éxito comercial, el F-8, también de 8 bits de la Fairchild, el PPS-8 de Rockwell y el PACE, de National Semiconductors, este último de 16 bits. Sin embargo, estos microprocesadores hasta el momento se habían considerado óptimos para el control de peque-

Año	Lenguaje		Uso
1954	Fortran	Fórmula translator	Científico
1956	Lisp	List processor	Inteligencia artificial
1959	Cobol	Common business oriented language	Comercial
1960	Algol	Algorihmic language	Científico
1962	APL	A programing language	Científico, modelos
1964	PL/I	Programing language I	Comercial, científico
1965	Basic	Beginner's all-purpose symbilic instruction code	Enseñanza
1971	Pascal	Blaise Pascal	Enseñanza sistemas

Lenguajes de programación

ñas máquinas automáticas. Nadie había pensado en incorporarlos a un ordenador.

El primer ordenador basado en la primitiva microplaqueta de la Intel fue el Altair 8800. La empresa que lo comercializó no se dedicaba directamente a ordenadores, sino a maquinaria de medida muy especializada. La MITS (Micro Instrumentation and Telemetry Systems) se decidió a fabricar un tipo muy especial de calculadores, y fue en ese momento cuando estudió las posibilidades de estas microplaquetas o microprocesadores. Su producto, el Altair 8800, no tuvo demasiado éxito. Sin embargo, fue el peldaño necesario para que aparecieran en el mercado muchísimas compañías que comenzaron a fabricar modelos de mini o microordenadores, saturando el mercado existente en aquellas fechas. El miniordenador (todavía no se utilizaba el nombre micro) empezó a estar de moda, aparecía en revistas especializadas, y era la panacea de todas las bondades. Las compañías intentaban captar clientes en todos los estratos de la capa social. Unas anunciaban su ordenador como óptimo para ciertas tareas empresariales, otros para aplicaciones típicas del hogar, o en tareas educacionales. Naturalmente, la aparición del software apropiado para estos tipos de máquinas tuvo una importancia capital en su difusión. Salvando excepciones muy honrosas, un médico no podría preocuparse en resolver sus problemas de historias clínicas, fichero de clientes, emisión y envío de facturas, etcétera, programando en COBOL, o en otro lenguaje exclusivo de profesionales. El apogeo del BASIC, del que van apareciendo sucesivas versiones, supuso una ayuda fundamental para la expansión del microordenador. Hoy en día los microordenadores son mucho más potentes que sus hermanos menores, y pueden programarse tanto en COBOL como en FORTRAN, o en otros lenguajes especializados. Pero es claro que el hombre común que adquiría un microordenador estaba interesado principalmente en resolver sus pequeños problemas cotidianos, y no tanto en aprender Informática.

Efectivamente, a partir de 1960 el software comienza a salir de un cierto letargo. Hasta el momento, es un software generalizado, que servía casi para cualquier aplicación, y que era vendido por la casa fabricante del ordenador correspondiente, incluyendo su precio en el de la máquina. Hasta ese momento, cada empresa que adquiría un ordenador partía del software incluido en su máquina, y desarrollaba un software específico a su medida. Nadie se preocupaba que ese software fuera "portátil", es decir, pudiera ser utilizado en otra compañía cuyas necesidades fueran comparables. A partir del año 60 comenzaron aparecer casas de software, cuya labor consistía en crear un software portátil y específico para cada actividad (comercial, de oficina, de gran número de actividades profesionales, etcétera). Estas empresas destruyeron el mito (bien alimentado por las casas fabricantes) de que el software que ellos ofrecían era inmejo-

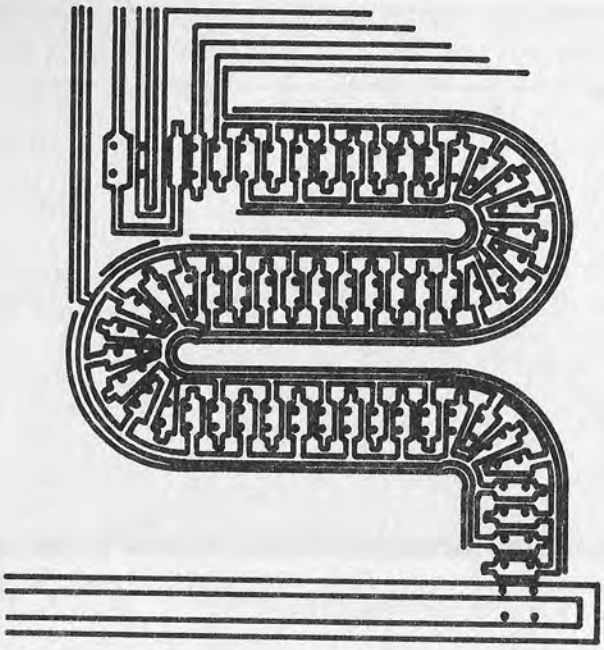


Fig. 2. Una de las primeras memorias de burbujas.

rable. Desde luego, se mejoró, e incluso se adaptó mucho mejor a cada necesidad de usuario, además de resultar "más barato" que el "gratuito" ofrecido por las grandes compañías fabricantes.

Otra decisión que ayudó en cierto modo al crecimiento de las casas de software fue la tomada por la Compañía IBM. A partir de 1969 se decidió a separar por un lado la facturación del coste de la máquina, y, por otro, el softwa-

Memoria de burbujas

Veamos sucintamente en qué consiste una memoria de burbujas. Las burbujas son pequeñísimos elementos cilíndricos (10^{-6}metros) que se mueven por una microplaqueta grande magnetizada de forma que los polos estén orientados en la orientación opuesta a las burbujas. Las memorias se mueven aplicando campos magnéticos externos.

Estas burbujas actúan como imanes dentro de un imán mayor que las contiene (y que suele ser de granate). Al ser magnetizables, su orientación es permanente, y así, son susceptibles de almacenar la información.

8008 ¿un producto rentable?

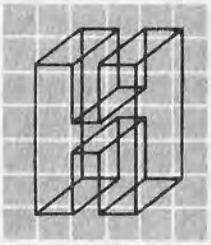
El 8008 fue el segundo procesador diseñado en los laboratorios de la Intel. Más potente que su predecesor, seguía siendo lento. No tuvo demasiado éxito comercial en sus comienzos, debido a que no era el óptimo para la aplicación específica para la que había sido diseñado. En efecto, el 8008 había nacido a partir de un contrato de la Intel con la compañía Display Terminals Corporation. Su finalidad era controlar una pantalla de visualización de rayos catódicos. Sin embargo, poco a poco, esta microplaqueta comenzó a utilizarse para muchas aplicaciones. Su juego de instrucciones era bastante mayor que el 4004, aunque para algunas pequeñas aplicaciones también resultaba escaso. Su memoria RAM (o de acceso directo) era de 256 bits. A partir de este momento, las memorias de toros de ferrita comienzan a ser abandonadas frente a estos nuevos sistemas de almacenamiento.

Además de todas esas ventajas, sobre todo para el usuario de pequeño nivel, el precio era también un aspecto importante a considerar. Para Intel fue un paso de prueba, que decidió a la Compañía a seguir en la misma línea, creando algo más tarde otro procesador más moderno y potente que el anterior: el 8080. Este nuevo microprocesador era sobre todo más rápido que el 8008 (de diez a cien veces más). Su memoria era mayor (de 16 a 65 Kbytes), y tenía casi el doble de instrucciones que el modelo anterior.

re, que hasta el momento la casa había ofrecido "gratuitamente". A partir de ese momento los usuarios se encontraron con otra libertad para seleccionar el software más acorde con sus necesidades. Nosotros no creemos que protestaran formalmente por haber perdido esa prerrogativa. Los gastos de un laboratorio de desarrollo de software de una gran multinacional son mucho más elevados que los de una pequeña compañía de software, especializada en usuarios concretos. Y éstos pueden obtener paquetes utilísimos a precios muy inferiores.

En los años setenta, las grandes compañías fabricantes de semiconductores se interesaron también en la producción de microordenadores. Estos se fueron construyendo cada vez con miras más altas. Se les incorporaron discos duros, impresoras sofisticadas, terminales muy especializados, etc. Se utilizaron en comunicaciones, difundiéndose en todo tipo de aplicaciones.

Interpretación de imágenes tomadas mediante radar



EMOS hablado en la Historia de la Informática del proyecto SAGE, norteamericano, y de otros proyectos para el Ministerio de Defensa de los Estados Unidos de América. Estos proyectos, desde el punto de vista técnico, están totalmente anticuados. Veamos, sin embargo, la misma aplicación, que tiene interés no sólo en el ámbito militar, sino en muchos otros campos, pero desde una visión absolutamente moderna, es decir, con los medios de que disponemos hoy en día, en 1987.

Consideramos, pues, la interpretación automática de mapas pluviométricos, digitalizados, creados con la información obtenida de radares terrestres o embarcados en satélites. La interpretación de imágenes por ordenador va teniendo cada día más interés, y ello por múltiples razones: cada día se utiliza más la información mediante imágenes, debido a que en un momento dado, una persona puede obtener mucha más información viendo una determinada imagen, que si se le va dando esa misma información escrita, por ejemplo. En ese momento concreto, cada persona selecciona inmediatamente aquella faceta o detalle de la imagen que le interesa, sin tener que ir secuencialmente despreciando las restantes. Al mismo tiempo, si la imagen es lo suficientemente útil, puede darle una idea de conjunto, de relación de unas variables con respecto a las otras, que los simples valores jamás pueden ofrecer.

Sin embargo, el proceso de interpretación de imágenes no sigue siempre la misma mecánica, no es el mismo, depende de la aplicación concreta, del entorno en que ésta se desarrolle y de los sensores utilizados. Es claro que, por ejemplo, el criterio de segmentación y clasificación puedan ser distintos, en dos aplicaciones de proceso de imágenes muy distintas. El esquema general puede ser el mismo, pero cada uno de los pasos a dar se hace teniendo en cuenta las características concretas de cada aplicación.

En las líneas siguientes vamos a ver un caso muy específico de enorme futuro y que representa una tecnología actualmente en ebullición: el estudio de la estructura espacial de mapas pluviométricos obtenidos mediante radar. El sistema utiliza ciertas técnicas de análisis de las imágenes obtenidas por radar, "basadas en el conocimiento", y constituye una herramienta muy importante para estudiar la distribución pluviométrica de determinadas zonas.

Los radares utilizados en Institutos Meteorológicos se usan entre otras cosas para obtener información sobre las intensidades pluviométricas de regiones geográficas bastante amplias. El estudio de este reparto espacial de la lluvia tiene múltiples aplicaciones. Así, en Agricultura, pueden hacerse estudios posteriores sobre cosechas, en Meteorología, son especialmente útiles para predicciones del tiempo a corto plazo, e incluso en Comunicaciones, el estudio puede tener interés a la hora de establecer redes de comunicación (momentos óptimos de transmisión, puntos críticos, etc.).

La información con la que se va a trabajar proviene de radares instalados en estaciones meteorológicas. Esta información consiste en "mapas pluviométricos de tierra" (tomados desde estaciones terrestres y no desde satélites) bidimensionales, de los que se desea obtener su estructura horizontal (estra-

tificada por capas). Los mapas se refieren siempre a pequeños intervalos de tiempo, a entornos de 5.000 a 10.000 kilómetros cuadrados, y para una altura de aproximadamente 1,5 kilómetros de la Tierra.

La tarea a realizar no es fácil. La clasificación de estos sistemas tiene que ser compleja, dada la enorme variedad y sofisticación de los sistemas de precipitaciones. Sin embargo, vamos aquí a revisar aspectos morfológicos (relativos a la forma de la imagen patrón que aparece en el mapa), además de otros aspectos intrínsecos, como la variación de las intensidades dentro del patrón. (Estudio de los niveles de gris.) Este último punto es muy importante en el establecimiento de redes de comunicación. En este caso, lo que interesa es la detección de ciertas formaciones que llamaríamos "peligrosas" para las comunicaciones por ondas electromagnéticas.

Como en otros análisis de imágenes, en este caso, la interpretación de cada imagen se realiza por un reconocimiento total o parcial de un conjunto de características específicas de las estructuras. La combinación de todas estas características (con su peso específico dentro de esta combinación) permitirán decidir cuál es la naturaleza de los objetos que originan esos patrones en la imagen. Por tanto, la tarea consiste en identificar el conjunto de características adecuado para la clasificación, asignar un conjunto de valores para cada características y definir un algoritmo o herramienta que combine estos datos, para los valores del patrón general. Por tanto, la interpretación de imágenes DI (imágenes digitalizadas) es un proceso que se realiza en dos etapas: la primera es una recopilación de pistas o huellas para la descripción total o parcial de las imágenes digitalizadas, y la segunda una clasificación de dichas DI (imágenes digitalizadas) mediante el adecuado proceso de decisión, que debe evaluar e interpretar la descripción anteriormente obtenida.

El propio proceso de decisión, en muchas ocasiones, sólo está definido parcialmente por la persona indicada, y aquí surge el problema, ya que el proceso de interpretación no será en estos casos transparente. Es por ello que el procedimiento de clasificación de imágenes debe tener en cuenta esta realidad.

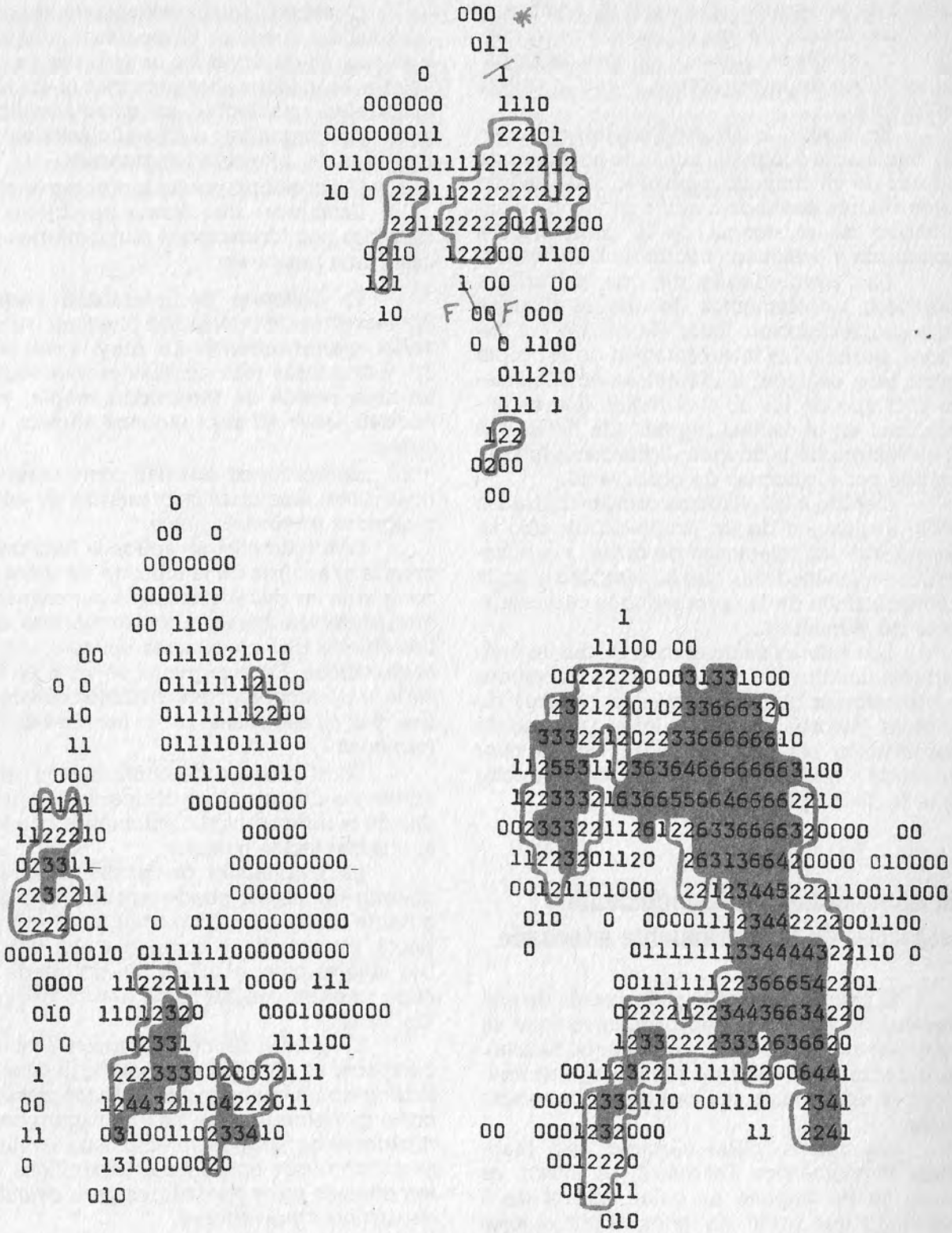
El procedimiento propiamente dicho comienza en una etapa de exploración por un intérprete humano, el cual deberá examinar el conjunto de imágenes digitalizadas, e interpretarlo *a priori*, es decir, describir y clasificar el conjunto de aquellas clases, atributos y características que utiliza él mismo cuando los procesos no son automáticos. A continuación deberá realizar un análisis preliminar del conjunto mediante otras herramientas, con el fin de definir procedimientos para la interpretación. De este análisis, se producirán un conjunto de descripciones y relaciones, cuyo resultado será un paquete de programas, que, a su vez, deberá ser probado con los datos preliminares, estudiando las posibles diferencias entre los datos obtenidos del ordenador por este sistema, y la interpretación *a priori*. Si la comparación no resulta satisfactoria, se repite el proceso completo de exploración, realizando naturalmente las oportunas modificaciones, hasta que se obtengan resultados aceptables.

La segunda etapa es el propio proceso de confirmación. En ella se utiliza un nuevo conjunto de datos no analizados previamente (pero naturalmente de la misma imagen) que es interpretado de nuevo tanto por la herramienta automática (programas) como por el intérprete humano, que, por otro lado, no debe conocer la estrategia específica utilizada en el proceso automático.

Si al comparar los resultados de las dos interpretaciones, el resultado es satisfactorio, las herramientas desarrolladas se aceptarán. En caso contrario, serán rechazadas, y se pasará a redefinir los criterios de decisión y las descripciones de la imagen. A partir de este momento se recomenzará una nueva exploración del mismo conjunto de imágenes, con un nuevo y más amplio criterio de reconocimiento de los datos.

Los mapas pluviométricos son imágenes digitalizadas (DI), es decir, un conjunto de valores que forman una matriz bidimensional cuyos elementos son números enteros dentro de una gama de valores definidos. Si los elementos de dicha matriz sólo pueden tomar como posibles valores ceros y unos, se dice que son imágenes digitalizadas binarias (en contraposición a las imágenes monocromas, que manejan niveles de gris).

Definamos a continuación claramente un elemento fundamental de la imagen: el pixel. Este nivel puede definirse por un conjunto de tres valores: dos de ellos son las coordenadas bidimensionales de su posición, y el tercero es un valor asociado, que tiene que ser un número entero. Otro elemento que también debemos definir es la "estructura", que es "un conjunto de pixels que satisfacen unas determinadas relaciones. Una característica es una



Ejemplos de clasificación de 3 objetos contenidos en diversas secciones de un GRM. (Mapa Terrestre Pluviométrico.)

estructura específica, que tiene un nombre, y una "descripción" de una estructura es un conjunto de valores asociados a algunas de las características de la estructura, o con la propia estructura.

En cuanto a las propiedades, pueden ser numéricas o lógicas, tomando un grupo de valores de un conjunto y estando cada uno de estos valores asociado a algún grado de plausibilidad de existencia de la característica examinada y designado por un símbolo distinto.

Las propiedades de una estructura dada son los elementos de que se dispone para su clasificación. Estos elementos, combinados, permiten la interpretación de la propia estructura, es decir, la identificación de la clase y el tipo de los objetos reales que se encuentran en la escena digitalizada de la que la estructura de la imagen digitalizada fue generada por el proceso de observación.

Debido a las distintas escalas utilizadas en la evaluación de las propiedades, sólo se consideran las relaciones de orden y equivalencia en las medidas que se efectúan y en la determinación de las propiedades características de la imagen.

Los valores se combinan mediante operadores multievaluados, es decir, funciones combinatorias que componen los valores de entrada (valores obtenidos en el proceso de observación por características) en un valor de salida o valor obtenido por el conjunto combinado de las propiedades observadas.

Adquisición del conocimiento y desarrollo de la herramienta adecuada

El primer paso es la observación de una muestra de imágenes representativa (que se debe llevar a cabo junto con el especialista), para definir las clases de patrones pluviométricos, y determinar sus características esenciales.

Los valores GRM (Ground Rain Map-Mapa Pluviométrico Terrestre) se filtran, es decir, se les impone un valor umbral de 5 mm/h, ya que están compuestos por objetos que no muestran ningún tipo de regularidad a primera vista. Su número varía desde uno a varias decenas por mapa, y su área puede variar desde un solo pixel hasta tomar el 25 o 30 % de la imagen completa. Su forma también puede ser cualquiera.

Considerando la distribución de grises, la forma del objeto en el espacio puede variar, y se podrán observar los objetos planos cuyos valores sean todos ellos próximos al umbral de visibilidad establecido, así como aquellos objetos que tengan uno o más núcleos dentro de regiones de intensidades menores.

Un prototipo puede ser el siguiente:

Establecer tres clases de objetos relacionados con formaciones pluviométricas típicas, como puede ser:

1) Sistemas de intensidad pequeña; 2) Sistemas de intensidad pluviométrica elevada, generalmente no muy extensos, y 3) Estructuras más amplias constituidas por un área básica de intensidad media, y que pueden tener además algunos núcleos especiales.

Se pueden tomar además como características discriminatorias la presencia de núcleos y algunos escalonamientos.

Con todo ello, se aplica la herramienta creada al análisis de la muestra de datos, y se comparan las dos versiones, la automática y la interpretación humana, observándose aquellos objetos que no concuerdan con las clases establecidas. En este punto, se ve si es necesario o no modificar los atributos seleccionados, y si es así, cuáles serán las nuevas clases formadas.

En el proceso de confirmación se examinan los objetos, y se comparan los resultados de la interpretación automática con los de la interpretación humana.

La evaluación de posibles escalonamientos del objeto puede establecerse del siguiente modo: muy poco compacta, muy compacta, intermedia y compacta con gránulos. Los objetos cuya clasificación no quede muy clara pueden considerarse dentro del grupo "dudosos".

El grado de consolidación (como de compacto es un objeto) se asigna, a éste, considerando un conjunto de atributos morfológicos o morfométricos, a los que llamaremos indicadores de escalonamiento. Para evaluarlos se utilizan unos operadores específicos. Existen además otros operadores para evaluación de distintos indicadores.

Si existen núcleos, se puede evaluar su intensidad dividiendo el objeto para dos valores distintos de intensidad pluviométrica. Una vez realizado, se vuelve a evaluar para cada nivel los núcleos y sus áreas. De este modo, cada objeto está marcado con el número y

descripción de los núcleos, en los dos niveles en que se ha dividido. En los diagramas, aparecerán el número de núcleos (uno, uno o más, o al menos dos; el tipo de núcleos y el nivel de intensidad (primer nivel o segundo nivel).

Este método de interpretación de las imágenes de radar ha sido probado en el norte de Italia, dando magníficos resultados, detectándose fácilmente formaciones especialmente peligrosas, e identificando y cuantificando estructuras de lluvia típicas. El sistema puede, sin embargo, mejorarse, sobre todo procurando hacer definiciones de clases que se aproximen más a la realidad, y analizando las estructuras formadas, que suelen originar una gran variedad de formaciones.

TERMINOLOGIA

Descriptores de la imagen. Características básicas del mapa pluviométrico extraídas de su imagen digitalizada y que identifican las estructuras significativas que aparecen en dicho mapa.

Estructura. En las imágenes digitalizadas correspondientes a los GRM (véase el término GRM), es un conjunto de pixels que satisfacen algunas relaciones concretas. Representa un sistema de precipitaciones, o al menos a un cúmulo nuboso significativo. Son estructuras los "gránulos", los "objeto binario", etc. (ver los dos términos entrecomillados).

Estructura horizontal. Referido a las formaciones de lluvia, indica la composición de las diferentes capas pluviométricas que se pueden considerar examinando la formación a diferentes alturas.

Gránulos. Se llama así a una estructura que puede aparecer en un mapa GRM, formada por un conjunto de pixels relacionados con uno determinado.

GRM. Ground Rain Map. Mapa terrestre de lluvia. Mapa pluviométrico obtenido para una altura de uno a uno y medio kilómetros.

Imagen binaria. Imagen digitalizada monocroma, con sólo dos niveles de gris: 0 (normalmente negro) para todos los pixels cuya intensidad luminosa no supera un umbral establecido, y 1 (blanco) para los que rebasan dicho umbral.

Imagen de niveles de gris. Imagen digitalizada, monocroma, en que se anota para cada pixel su intensidad luminosa dentro de una gama cuantificada formada por ocho niveles (si el sistema representa cada pixel mediante tres bits), dieciséis niveles..., hasta 256 niveles (ocho bits) o más, en los sistemas profesionales de alta precisión.

Imágenes digitalizadas. Representación numérica en forma matricial obtenida a partir de una imagen continua (analógica) facilitada por una cámara de vídeo, de infrarrojos, etc., un sistema de radar... o cualquier otro sensor. Cada unidad de imagen (pixel) se convierte en la digitalización en una pareja de elementos numéricos: el primero de ellos, formado por dos valores, indica las coordenadas del pixel correspondiente, dentro de la matriz de puntos en que se convierte la imagen; el segundo elemento define la intensidad luminosa de la unidad de imagen de que se trata, bien como tres valores (correspondientes al rojo, verde y azul) en los sistemas en color, bien como un número representativo del nivel de gris en los sistemas monocromos. La imagen digitalizada monocroma puede ser binaria o de niveles de gris.

ISIID. Interactive System for Image Interpretation Definition. Conjunto de herramientas diseñadas (Cugini, 1983) específicamente para el análisis y proceso de imágenes obtenidas mediante radar, que se han convertido prácticamente en un estándar.

Mapa pluviométrico. Mapa representativo de la situación atmosférica a una altura determinada, que refleja las distintas formaciones de lluvia, para una zona acotada. Sirve para predicciones de tiempo a corto y medio plazo, y otras determinaciones del estado de la atmósfera. Son utilizados en determinaciones meteorológicas, estableci-

miento de condiciones de transmisión para redes hertzianas y de comunicaciones, etc.

Objeto binario. Estructura compuesta de gránulos, que satisface algunas relaciones.

Sistemas de precipitaciones. Acúmulos nubosos cuya composición y estructura puede determinar el nivel y la distribución de las precipitaciones. Los mapas pluviométricos reflejan los sistemas de precipitaciones presentes en el área a la que se refiere el mapa.

Técnicas basadas en el conocimiento. Este término en el ámbito de la interpretación de imágenes alude a los procesos que incluyen técnicas de razonamiento aproximado, informaciones de tipo heurístico, etc., en contraposición a los procesos de tipo algorítmico cerrados y perfectamente definidos. Es usual incluir en un sistema de interpretación de imágenes de radar (y procesamiento de estas informaciones para predicciones), sistemas expertos que realicen un procesamiento inteligente de las informaciones obtenidas mediante los algoritmos estándar predefinidos, tan conocidos.

Umbral de binarización. Valor de gris que se utiliza como barrera o límite para dividir los pixels según su intensidad luminosa en dos únicos grupos con el fin de simplificar la imagen. A los pixels que están por debajo del umbral se les atribuye valor cero en la imagen binaria, y a los que están por encima, valor uno.